# Comparative Framework for Service Reliability in Electric Distribution Systems

By

Doeg Rodríguez Sanabria

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

MASTER OF SCIENCE
In
ELECTRICAL ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
2005

Approved by:

| | |
|---|---|
| José R. Cedeño Maldonado, PhD<br>Member, Graduate Committee | Date |
| Alberto Ramírez Orquín, PhD<br>Member, Graduate Committee | Date |
| Efraín O'Neill Carrillo, PhD<br>President, Graduate Committee | Date |
| Francisco Maldonado Fortunet, PhD<br>Representative of Graduate Studies | Date |
| Isidoro Couvertier, PhD<br>Chairperson, Electrical and Computer Engineering Department | Date |

# ABSTRACT

Reliability indices are becoming the cornerstone for a liberalized energy market, and the main component of performance based rates (PBRs). Yet, comparison of reliability indices from intrinsically different systems provides arguable results. A method producing coherent, comparable reliability numbers was sought after. The framework would be used to compare reliability index results from intrinsically different distribution systems. A theoretical foundation for a comparative framework for electric distribution service indices was developed. Reliability indices were calculated based on the current standards used by the industry for qualifying service. A mathematical model for the framework was proposed, and several implementation methods considered. General implementation guidelines for the model are included in the document. Actual field data from varying utilities were used for the completion of this project. Of the field data, fault cause and outage analysis was completed. Additionally, these results were compared to those of previous studies on the subject of outage occurrences and fault causes. Indices for unique systems or system areas with and without the use of the framework were obtained. Results obtained from application of the framework to actual distribution systems are presented in the document. The scope of the project was seven years, for which actual system data was made available.

# RESUMEN

El estudio de confiabilidad de servicio de distribución eléctrica se esta transformando en la base de un mercado liberalizado de energía. Aun así, la comparación de resultados de sistemas con características diferentes produce un análisis cuestionable. Se quería obtener un método para coherentemente comparar índices de confiabilidad. Éste seria utilizado para comparar índices de sistemas de distribución intrínsicamente diferentes. Para obtener resultados, se utilizaron índices estándares de esta industria. En este proyecto, se desarrolló la base teórica de un marco para la comparación de índices de confiabilidad para sistemas de distribución eléctrica. Un modelo matemático para el marco de comparación fue propuesto y varias implementaciones de éste fueron consideradas. Guías generales para la implementación de este modelo se incluyen en el documento. Datos reales de diferentes sistemas fueron utilizados en el proyecto. Se completo un análisis de causas de fallas y eventos de interrupción de los datos obtenidos. En adición los resultados de este análisis se compararon a resultados de trabajos anteriores en el área de estudio. Se obtuvieron índices de confiabilidad para sistemas únicos o áreas de un sistema con y sin el uso del marco de comparación. Resultados obtenidos utilizando el marco de comparación son presentados en el documento. El proyecto tenia un alcance de siete años, para los cuales se obtuvo datos reales de sistemas de distribución.

To things that could not be.

# ACKNOWLEDGEMENTS

I would like to dedicate this section to present my appreciation to all the people that made this project possible. First I must thank my project advisor, Efraín O'Neill-Carrillo, for his patience with project discussions and innumerable diverging strategies. Without his help, this project would have not been completed. Gratitude is also extended to my graduate committee, José R. Cedeño-Maldonado and Alberto Ramírez Orquin, for their involvement during the course of this ordeal.

I also greatly appreciate of the help provided by Thomas S. Key and Arshad Mansoor from EPRI-Solutions Corporation. Their input and original ideas became the basis for the project and was invaluable towards its completion. Without their involvement, this project would have taken a very different approach, if any.

Last but not least, I thank my family for their love and support during these difficult times. I would also like to extend a note of gratitude to all my friends and their support during the project. I must thank Carlos M. Torres and Carlos A. Vega for their input on the art of constructing a thesis and Angel Rivera for providing his insight on serious topics during the project. Last but definitely not least, I must thank Brenda Liz Pérez for her continuing support and friendship, even in the darkest of times.

# TABLE OF CONTENTS

**Page**


ABSTRACT ..... ii
RESUMEN ..... iii
ACKNOWLEDGEMENTS ..... vi
TABLE OF CONTENTS ..... vii
LIST OF TABLES ..... ix
LIST OF FIGURES ..... x
CHAPTER 1 Introduction ..... 1
1.1 Objectives ..... 2
1.2 Outline of the Thesis ..... 2
CHAPTER 2 Literature Review ..... 3
2.1 Power Distribution Systems ..... 3
2.1.1 Overview ..... 3
2.1.2 Distribution Substations ..... 4
2.1.3 Distribution Feeders ..... 7
2.1.3.1 Radial Feeders ..... 9
2.1.3.2 Network Feeders ..... 11
2.2 Power Quality Indices ..... 12
2.3 Distribution Reliability / Performance Indices ..... 18
2.3.1 Reliability Index Tracking ..... 20
CHAPTER 3 Electric Power Distribution Reliability Indices ..... 23
3.1 Statistics Theory ..... 23
3.1.1 Probability Distributions ..... 23
3.1.1.1 The Normal Distribution ..... 26
3.1.1.2 The Log-Normal Distribution ..... 27
3.1.2 Confidence Intervals ..... 29
3.2 IEEE 1366 Distribution Reliability Indices ..... 31
3.2.1 Sustained Interruption Indices ..... 33
3.2.1.1 Sustained Average Interruption Frequency Index (SAIFI) ..... 33
3.2.1.2 Sustained Average Interruption Duration Index (SAIDI) ..... 34
3.2.1.3 Customer Average Interruption Duration Index (CAIDI) ..... 34
3.2.1.4 Customer Total Average Interruption Duration Index (CTAIDI) ..... 35
3.2.1.5 Customer Average Interruption Frequency Index (CAIFI) ..... 35
3.2.1.6 Average Service Availability Index (ASAI) ..... 36
3.2.1.7 Customers Experiencing Multiple Interruptions (CEMIn) ..... 36
3.2.2 Load Based Interruption Indices ..... 36
3.2.2.1 Average System Interruption Frequency Index (ASIFI) ..... 37
3.2.2.2 Average System Interruption Duration Index (ASIDI) ..... 37
3.2.3 Momentary Interruption Indices ..... 37
3.2.3.1 Momentary Average Interruption Frequency Index (MAIFI) ..... 38
3.2.3.2 Momentary Average Interruption Event Frequency Index (MAIFIE) ..... 38
3.2.3.3 Customers Experiencing Multiple Sustained Interruption and Momentary Interruption Events (CEMSMIn) ..... 38
3.2.4 Major Event Days ..... 39

3.3 Factors Affecting Reliability ..... 41
3.3.1 Interruption and Fault Types ..... 43
3.3.2 Factors Affecting Reliability ..... 44
3.4 Comparative Framework ..... 47
3.4.1 Objectives of framework ..... 47
3.4.2 Proposed Method ..... 48
CHAPTER 4 Comparison Framework Implementation ..... 51
4.1 Data Verification ..... 51
4.2 SAIDI/SAIFI ..... 54
4.3 MED Threshold ..... 55
4.4 Fault Code data ..... 56
4.5 Weighting Process ..... 57
4.5.1 Weighting Process Phase 1 ..... 58
4.5.2 Weighting Process Phase 2 ..... 58
4.5.2.1 Mode 1 PAIR ..... 59
4.5.2.2 Mode 2 MEAN ..... 60
4.5.2.3 Mode 3 MAXIMUM ..... 60
4.5.3 Weighting Process Phase 3 ..... 60
4.6 Average Daily Normalized SAIDI ..... 62
CHAPTER 5 Discussion ..... 64
5.1 Reliability Index Calculation ..... 64
5.2 Fault code use and impact ..... 64
5.3 Framework ..... 67
5.3.1 Framework RIPPER results ..... 67
5.3.2 Major Events ..... 73
5.3.3 Framework WEIGHTER results ..... 75
5.3.4 Framework STRIPPER Results ..... 78
5.4 Other Results ..... 86
CHAPTER 6 Conclusions and Future Work ..... 87
6.1 Conclusions ..... 87
6.2 Future Work ..... 89
REFERENCES ..... 90
APPENDICES ..... 95

# LIST OF TABLES

**Table** **Page**

Table 2.1.1. Distribution Voltage Classes ..... 5
Table 2.2.1. Power Quality Indices ..... 16
Table 2.3.1. Reliability Index Use ..... 20
Table 2.3.2. SAIDI/SAIFI Values for Several Surveys ..... 21
Table 2.3.3. SAIDI Values for Several Utilities ..... 21
Table 2.3.4. The Service Reliability by Number of Nines ..... 22
Table 3.1.1. Probability for Typical Measurements ..... 30
Table 3.1.2. Confidence Interval for Selected Probabilities ..... 30
Table 3.1.3. Lognormal Confidence Bounds ..... 31
Table 3.2.1. Factors for calculation of IEEE1366 Indices ..... 33
Table 3.3.1. Equipment included in calculations ..... 42
Table 3.3.2. Fault Causes by Type ..... 44
Table 3.3.3. Fault Causes ..... 44
Table 3.3.4. Number of Phases Involved in Fault ..... 45
Table 3.3.5. Reliability of Network Topologies ..... 46
Table 3.3.6. SAIDI Values for various Australian Service Territories ..... 46
Table 5.2.1. Equipment failure percentages ..... 66
Table 5.3.1. ANDS results: All Areas without adjustment or compensation ..... 69
Table 5.3.2. ANDS results for All Areas; MED Compensated ..... 73
Table 5.3.3. Fault Code Weights ..... 76
Table 5.3.4. ANDS Variance: Adjusted/No Compensation ..... 82
Table 5.3.5. ANDS Variance: Adjusted/Compensated ..... 83

# LIST OF FIGURES

**Figure** **Page**


Figure 2.1.1. Power System Schematic ..... 4
Figure 2.1.2. Sample Distribution Substation ..... 4
Figure 2.1.3. Breaker Configurations ..... 7
Figure 2.1.4. Radial Circuits ..... 9
Figure 2.1.5. Primary Loop and Primary/Secondary Selective Examples ..... 10
Figure 2.1.6. Grid and Spot Network Examples ..... 12
Figure 3.1.1. Distribution Skew Examples ..... 24
Figure 3.1.2. Normal Distribution PDF & CDF ..... 27
Figure 3.1.3. Lognormal Distribution PDF & CDF ..... 29
Figure 4.1.1. Lognormality test results ..... 53
Figure 4.5.1. Framework Process Flowchart ..... 61
Figure 5.2.1. Fault Causes by Percentage ..... 65
Figure 5.2.2. Faults by Feeder Client Base Affected ..... 66
Figure 5.3.1. ANDS values: No Adjustment or Compensation ..... 69
Figure 5.3.2. Yearly SAIDI Variation: Lima Area ..... 70
Figure 5.3.3. Yearly SAIDI Variation: Charlie & Lima Areas ..... 71
Figure 5.3.4. Yearly MED Variation: All Areas ..... 72
Figure 5.3.5. Yearly SAIDI variation: All Areas ..... 72
Figure 5.3.6. ANDS: All Areas Normal/Compensated Comparison ..... 73
Figure 5.3.7. Yearly SAIDI: All Areas Normal/Compensated Comparison ..... 74
Figure 5.3.8. Yearly SAIFI: All Areas Normal/Compensated Comparison ..... 75
Figure 5.3.9. Final ADJUSTER Weights: Charlie & Lima Areas ..... 77
Figure 5.3.10. ANDS: All Areas Adjustment/No Compensation Comparison ..... 78
Figure 5.3.11. Yearly SAIDI: Hotel Area Normal/Adjusted Comparison ..... 79
Figure 5.3.12. Stack Diagram ANDS ..... 79
Figure 5.3.13. Yearly SAIDI Variation: All Areas Adjusted/Compensated Comparison ..... 80
Figure 5.3.14. ANDS Variance: All Areas Adjusted/No Compensation Comparison ..... 81
Figure 5.3.15. ANDS Variance: All Areas Adjusted/Compensated Comparison ..... 83
Figure 5.3.16. Fault Profile: Golf Area ..... 84
Figure 5.3.17. Fault Profile: Hotel Area ..... 85

# CHAPTER 1 INTRODUCTION

As deregulation of the electric utility sector dissolves old monopolies and operation practices, new market strategies are being created to facilitate the operation of electric grids around the world. Electric utilities now face a liberated energy market, where they compete with other utilities for client loads. Furthermore, the introduction of new technologies provide for impact on service quality and reliability that had not been considered in the past. Service quality in addition to service reliability has also become an important market strategy, due to the proliferation of sensitive customer loads. As this new energy market matured, a need to quantify and qualify the reliability and quality of service grew.

For the purpose of quantifying and describing utility service, regulatory agencies started compiling service reliability data from utilities. As the need for this data increased, standardized methods were sought to provide for compatible numbers between utilities and regulatory agencies. As a result, the IEEE began work on a standardized approach for reliability index calculation with the first draft of IEEE Standard 1366 [1]. Approved in 2001 and updated in 2004, Standard 1366, "Guide for Electric Power Distribution Reliability Indices", provides a common procedure to obtain utility reliability indices from different utilities. Analogous to Standard 1366, power quality applications refer to IEEE Standard 1159-1995 [2], "Recommended Practice for Monitoring Electric Power Quality".

Widespread use of the indices, indicate that there might not be a direct equivalence between results of reliability indices for different utilities [3]. Additionally, there are no clear goal values for utilities to seek, since reliable service for a specific utility will depend on the system's topology and inherent characteristics. In other words, direct comparison of reliability indices can be completed when system characteristic variables are, in a sense, normalized for comparison. Utility context information is typically sacrificed for the sake of simplicity in index calculation. Nevertheless, it is this information that might invalidate the usefulness of the results, because comparisons of fundamentally different things provide arguable results.

The main goal of this work is to provide the theoretical basis to provide a normalized approach or framework for reliability index comparison. In this approach, topology and operational differences of systems will be taken into account when indices are compared and calculated, to provide for coherent values with different systems. Possible integration of these findings with power quality indices will also be considered.

## 1.1 Objectives

The common framework concept is based on the need to have performance based electrical distribution service indices which are unbiased towards distribution system uniqueness. To accomplish this task, the following objectives were proposed:

- Distribution interruption data will be obtained, and a database will be created with this information.
- Fault causes and codes for the utilities will be studied and added to the database.
- Feeder data will be complemented with geography/topology information.
- Possible methods to relate existing power quality indices to power distribution reliability indices will be studied.
- A mathematical process will be proposed to adjust standard reliability indices for improved comparison of different distribution systems.

## 1.2 Outline of the Thesis

This document consists of six chapters. Chapter 1 is this introductory section. Chapter 2 provides a literary review of completed work in the related areas of research. Chapter 3 gives detail of the work to be completed and in-depth information on the processes and theory behind the project. Chapter 4 provides insight on the implementation of the methods proposed on the previous chapter. Chapter 5 provides the discussion of results. Chapter 6 presents the conclusions and future work for the project.

# CHAPTER 2 LITERATURE REVIEW

## *2.1* ***Power Distribution Systems***

Electric power distribution systems are the most inglorious part of modern power systems, but their function is essential; they connect utilities' biggest customer base to their systems. The proliferation of modern consumer electronic equipment has been, in part, a response to cheap, readily available and reliable electric service to millions (upon millions) of domestic customers. It is not to be misunderstood that distribution systems only service domestic clients, but they are a large portion. Smaller industrial and commercial customers also are connected to distribution systems, but in much lower numbers than domestic clients.

### 2.1.1 Overview

A distribution system is generally considered to be composed of all the lines and equipment that transport electric power to customers from a source, generally considered the local substation –everything from the step down substation to the client entry-point. Figure 2.1.1 presents a very simple, conceptual diagram of a power system. The last two elements, substations and feeders, are included in what is the utility's distribution network. Once power is transmitted through the transmission (745-115kV), and sub-transmission (69-35kV) systems it is, once again, stepped down (reduced in voltage) at distribution substations, and sent to loads through lower voltage lines (35-5kV) or feeders which serve the clients through distribution transformers (120/240 or 277/480V).

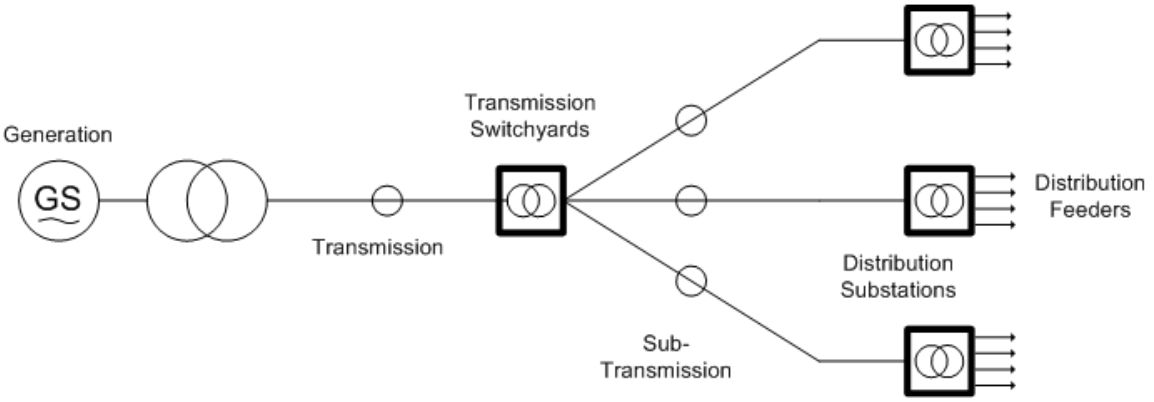



**Figure 2.1.1. Power System Schematic**

## 2.1.2 Distribution Substations

Figure 2.1.2 illustrates a conceptual, albeit simple, distribution substation. The substation is powered by the sub-transmission or transmission network from the utility. This is usually done at subtransmission levels of 35-69 kV even though exceptions to this rule exist [4][5]. Otherwise it might be connected directly to the transmission system at voltages of 115 kV or higher.

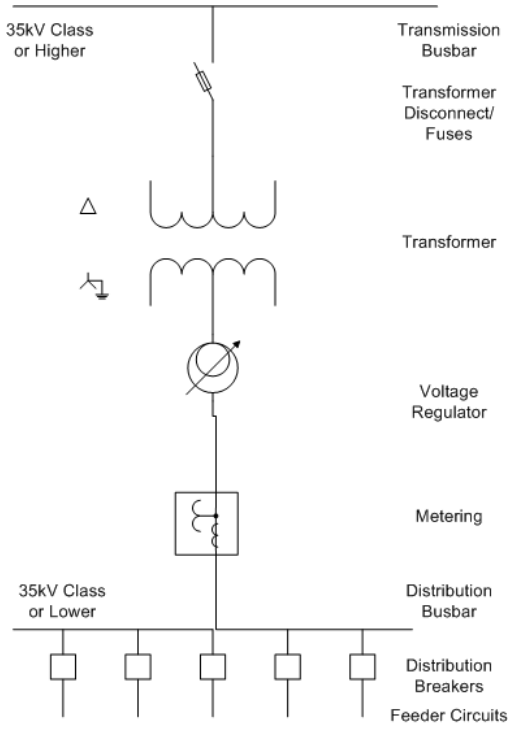



**Figure 2.1.2. Sample Distribution Substation**

The transmission voltage node or *bus* feeds the substation transformer though some kind of disconnecting device. This device might be a load interrupting breaker, or a disconnect switch with fuses. In many substations, breakers are installed on both high and low voltage sides of the transformer. The fuse or breaker on the high side act as a backup protection for feeders in case their own protection does not act, or fails. Primarily though, the high side protection is there to protect the substation transformer which is, by far, the most expensive component of the substation.

The transformer is a step down type unit. It provides distribution class voltage by bringing down sub-transmission voltages. Distribution level voltages are normally in the 15kV class (2-15kV) but it is not uncommon for utilities to use 25kV or 35kV class distribution equipment [4][5][6]. Typical distribution voltages are presented in Table 2.1.1.

**Table 2.1.1. Distribution Voltage Classes**

| **Distribution Voltage Classes and Levels** | |
|---|---|
| **Voltage Class** | **Common Voltage Level** |
| *5kV* | *4.16kV*<br>*4.8kV* |
| *15kV* | *12.47kV*<br>*13.2kV*<br>*13.8kV* |
| *25kV* | *24.94kV* |
| *35kV* | *34.5kV* |

*Source: Reference [4]*

The transformer typically is fitted with a load tap changer (LTC) to help with voltage regulation. The tap changer helps by changing the transformation ratio and allowing voltages to go above or below nominal or design values in the secondary side. Additional to the LTC on the secondary side, most transformers are fitted with manual taps on the high side, to compensate for line voltage drop. These fixed taps alter ratios by 5% in any direction. In some substations, instead of having one regulator for the secondary bus, each feeder is fitted with an exclusive one, but this can be an expensive proposition and used only in critical cases.

Protection for the distribution feeders are generally provided by breakers. This can be of the vacuum or inert gas (SF6) types, even though older technologies are still in use, such as oil or air blast breakers. At the distribution level, breakers are generally called reclosers, due to their operational characteristics. Reclosers are named so because they trip and close the circuit automatically. They normally trip instantly, and close the circuit automatically after a certain delay. This is done since most faults on feeders are temporary, and it is unnecessary to force long interruptions if the fault is not persistent. In case the fault persists, the protection equipment locks out the recloser so that crews may go in and inspect the problematic feeder.

A final component of a distribution substation is the metering system. Utilities measure customer demand on a substation for a myriad of reasons. They meter to keep track of power flow in the area, and for planning purposes. Additionally they serve as a check for individual meters at the clients' entrances, and can be used to account for distribution losses or energy theft.

Distribution networks can vary in topology, but are mostly radial systems. The use of radial systems is due to economics and operational efficiency. These are easier to operate and maintain, and protection systems are simple and cheap. Protection in radial systems is simple due to the fact that power flow occurs in only one direction, thus fuse links can be used for most applications, and no complex protection schemes or networks are required. Nevertheless, there are some other strategies used, for both the secondary (load) and primary (source) sides in the substation and network. Several configurations can be used on the high and low sides of the substation [7].

On the high side, for more critical substations, breaker-and-half, ring or multiple feed bus configurations be used. These configurations are typical in important bulk transmission substations or switchyards, because they provide redundancy. The redundancy allows for maintenance personnel to work without de-energizing the entire bus, causing no down-time. Also, these configurations are always fed from multiple locations, which provides for protection from downed lines. The following illustrations present multiple feed configurations. The breaker conditions can be selected to be

normally open or closed depending on the application. Several typical breaker configurations are presented in Figure 2.1.3.

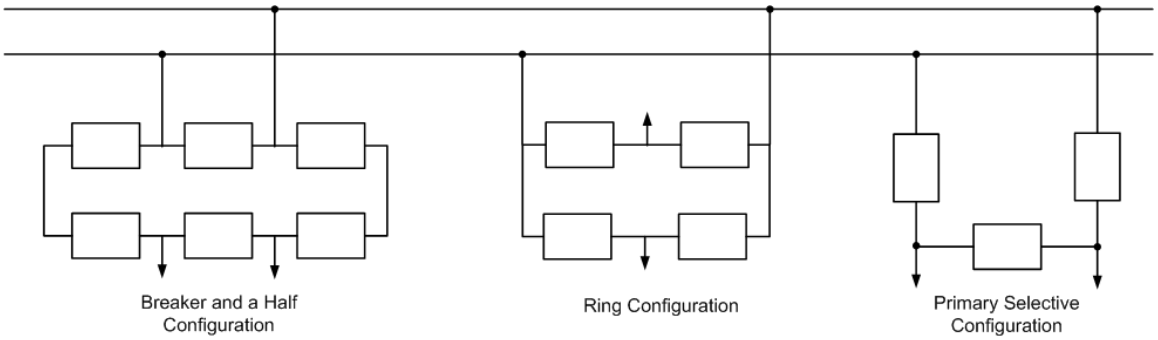



**Figure 2.1.3. Breaker Configurations**

Redundancy can also be added to the substation by using more than one transformer. Two or more transformers can be used to serve the load, by using load balancing between them. In case one fails, the other can carry full load. To do this, transformers must be selected for capacity in excess of total load. This method also extends the life of the transformers by having them operate below nominal (design) capacity, which minimizes equipment wear. Many more reliability considerations for distribution systems are presented in the IEEE493-1997 Standard [8], "Recommended Practice for the Design of Reliable Industrial and Commercial Power Systems", along with expected failure rates for virtually all kinds of equipment.

### 2.1.3 Distribution Feeders

Feeder is the name given to distribution lines that carry power throughout the distribution system. Radial distribution feeders are characterized by having only one path for power to flow from the source (substation) to each customer load. The distribution network is composed of the substations and the feeders that they supply. Typical components of the network are the following:

- Three-phase primary *mainlines*
- Three-phase, two-phase and single phase *laterals*
- Voltage regulators
- In-line transformers

- Shunt capacitor banks
- Distribution transformers
- *Secondaries*
- Three-phase, two-phase and single-phase loads

Distribution feeders are inherently unbalanced due to the proliferation of single-phase loads. That is, although ideally one wishes to have the same amount of power being delivered on all three phases of any distribution system, this cannot be achieved due to the characteristics and operation of customer equipment connected in each phase. Unequal loading is not the only cause of system unbalances, non-equilateral conductor spacing of three phase line segments, overhead or underground, also introduce unbalance into the system [9].

Feeders are composed of sections – each serving a purpose. The *mainline* is the backbone of the feeder and is typically a three-phase line generally designed for 400A nominal and 600A contingency loading. From this mainline, three-phase, two-phase or single-phase *laterals* extend. Laterals are branches that are connected to the main line to serve loads, which can have laterals of their own. To protect mainlines from faults in the laterals, these are typically fused or switched by remote reclosers.

The most common configuration for feeders is a four-wire wye configuration; one conductor for each of the three phases plus a solid multi-grounded neutral conductor. It is called a wye because all three phases share a common point of connection, and the topology resembles a letter Y. Nevertheless, less common configurations are three wire wye and delta configurations. Four-wire wye configurations are extensively used for their safety and easy fault protection [4]-[7]. This circuit type permits the use of under-rated voltage equipment by using phase-to-ground voltages, and fuse protection for ground faults.

Distribution systems can have many arrangements, with variations of both primary and secondary configurations [4]-[6]. Some configurations are:

- **Radial**
  - Radial with tie points
  - Primary loop

- Primary/Secondary selective

- **Networked**
  - Spot
  - Grid

Radial circuits have the following advantages over networked circuits:

- Easier fault current protection
- Lower fault currents
- Easier voltage control
- Easier prediction and control of power flows
- Lower cost

Even though radial circuits are simpler to operate and maintain, in order to increase the reliability and power quality of distribution feeders, networked configurations must be used [4][5][8]. Both types are discussed briefly in the following sections.

#### *2.1.3.1 Radial Feeders*

Radial feeders are normally provided with tie points between adjacent feeders. These are operated manually in most cases, but can be instrumental in the restoration of service after sectionalization has taken place. An illustration presents this – Figure 2.1.4.

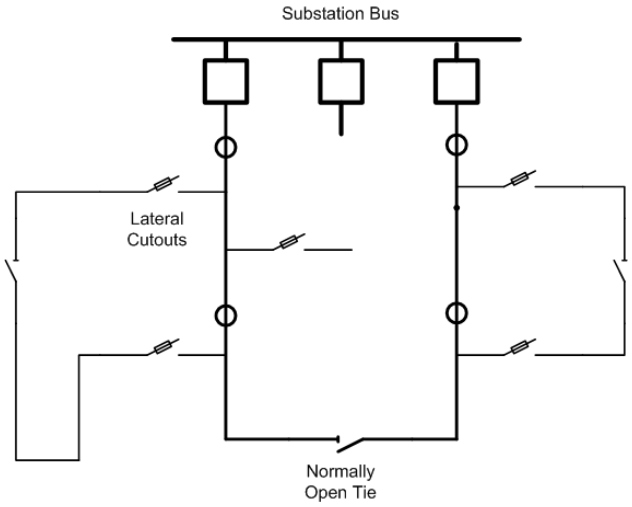



Figure 2.1.4. Radial Circuits

For radial feeders, two higher reliability configurations are most popular. Presented in Figure 2.1.4, these are the *primary loop* and the *primary* or *secondary selective*. Figure 2.1.5A presents a sample configuration for a primary loop system. The primary loop consists of two feeders which serve one or several loads. If one feeder fails, the normally open switch can be automatically (or manually) closed to feed the loads from the backup feeder, once the bad section has been sectionalized. Many underground systems are operated in this way, since repairs to underground cables are complex operations and would create long interruptions. By reconfiguring transformer switches, customers can be reconnected to a good feeder while the failed one is in repair, decreasing downtime considerably. Loops are normally operated in open-loop mode, not to affect installed protection. Nevertheless, after serious modification some utilities operate primary loops in closed-loop mode [4]. This type of open-loop scheme provides no protection from momentary interruptions or voltage sags.

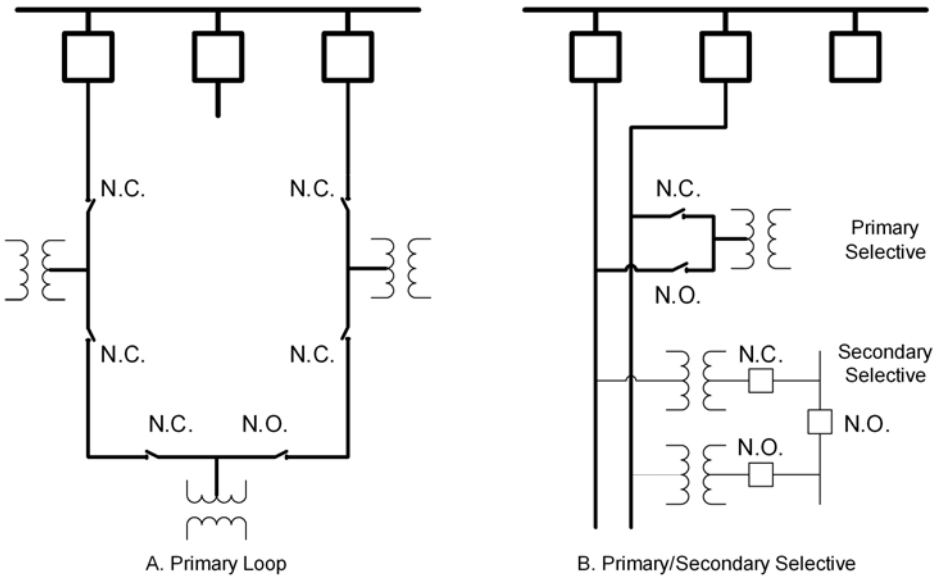



**Figure 2.1.5. Primary Loop and Primary/Secondary Selective Examples**

Primary and secondary selective are another type of radial distribution circuits with improved reliability (Figure 2.1.5B). In the primary selective scheme, two feeds from different circuits are available for a customer load. The load is fed from one circuit at a time. In case this circuit fails, the system can automatically switch to the other

circuit. In the secondary selective scheme, the operation is the same, but the switching is done in the secondary. Of these two, primary selective is used more commonly since the expense of an additional backup transformer is prevented. This scheme can help with momentary interruptions and sags if static switches are used to transfer the load in less than a half cycle when necessary.

#### *2.1.3.2 Network Feeders*

Not all distribution networks are radial, *spot* and *grid* secondary schemes are networked (see Figure 2.1.6). These schemes create a networked feed at the secondary level; the primary level is still radial (which can be any of the mentioned in section 2.1.3.1). Spot and grid networks are similar but differ in one characteristic: a spot network feeds a building or complex, while a grid network supplies several buildings in a larger area. Spot and grid networks have secondary circuits fed from several primary feeds, and they are all normally closed. Thus if one of the primary feeds fail, the loads continue to be fed from any of the other primary feeds trough the secondary network. Network protectors prevent backfeed from the secondary network to a fault in any of the primary circuits by using reverse power protection; this severs the spot or grid connection to the failed primary feed. Secondary grid networks can have peak loads of 5 to 50 MVA, but capacities of 250MVA are not unheard of [4]. Grid networks can supply both three-phase and single-phase loads.

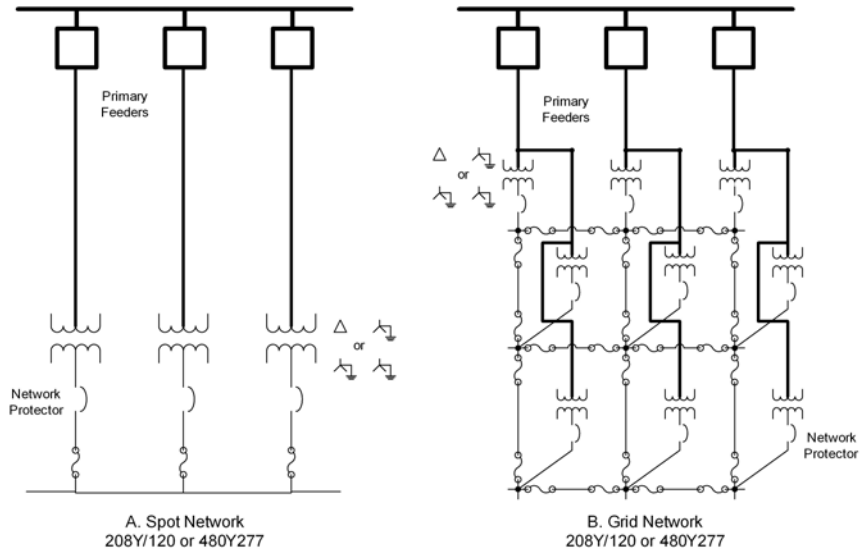



**Figure 2.1.6. Grid and Spot Network Examples**

Secondary networks are as a rule, fed from the same substation bus. This reduces or eliminates circulating currents and gives better load division and distribution among circuits. Circulating currents can trip network protectors; this problem is reduced by using the same bus. Still, for complete redundancy, networks can be fed from distinct substations.

## 2.2 *Power Quality Overview*

Electric energy demand is constantly increasing and with it, a growing expectance of high reliability and quality of service. Recent technological developments have increased the propagation of sensitive loads throughout the grid, which make customers more sensitive to service quality. As a result, customers have become less tolerant to electric energy service of low quality. The necessity for better service has prompted an increased interest in the field of study generally called, *power quality*. Power quality has various definitions and applications; subsequently, means many different things to many people. Nonetheless, a widely accepted definition is: "the measure, analysis, and improvement of bus voltage, usually a load bus voltage, to maintain that voltage to be a sinusoid at rated voltage and frequency" [9]. In other words, any event that might affect the expected reliability or quality of electric power can be considered a power quality

problem. Power quality phenomena can be classified in two general types: non-synchronous and synchronous. Synchronous phenomena are described as those events that affect power quality in a way that is synchronized or aligned with the system frequency. Conversely, non-synchronous phenomena have no distinct relation to the system frequency. Some non-synchronous phenomena that affect power quality are the following:

- noise
- asynchronous impulses and notches
- asynchronous sinusoidal waveforms
- flicker
- inrush current
- geomagnetic disturbances

Synchronous or periodic phenomena can include, but are not limited to:

- harmonic signals
- synchronous impulses and notches
- magnetizing currents

All these events can cause a load to become unserved or cause misoperation due to service inadequacy. Any of these events may not only prevent adequate operation of a device, but may also cause permanent damage, which is undesirable for the utility and the client. Since any of these events alter the specified electric service being delivered to the load, they are all considered power quality problems and should be prevented. The most common events that affect power quality are sags and swells, spikes and notches and harmonic pollution. Most power quality problems experienced on power systems are based on these types of events.

Harmonic pollution, voltage sags/swells and spikes/notches are the primary sources for power quality problems and as such, will be more extensively discussed in this review. Harmonic contamination is one of the earliest known power quality problems, encountered as early as 1893 [10]. Harmonics are signals existing in a power system with a frequency other than the fundamental system frequency. They can be classified in two main categories: characteristic and non-characteristic. Characteristic

harmonics are those that are multiples of the system fundamental frequency and are very common in power systems. Non-characteristic harmonics on the other hand, are those that are not a whole multiple of the fundamental frequency. These signals of intermediate frequencies are referred to as inter-harmonics. Inter-harmonics are named this way due to their existence between the whole multiples of the system fundamental frequency [9]. Other non-characteristic harmonics are sub-harmonics which are composed by frequencies lower than the fundamental frequency and can be the result of ferro-resonance or machine vibrations [9][11]. Non-characteristic harmonics can also be the result of unbalances in the power system. The asymmetrical and un-transposed installation of lines can cause varying mutual inductances between phases which can produce some of this non-multiple frequencies [9][11]. Non-characteristic harmonics are typically not generated by loads and many are due to uncontrollable phenomena, in contrast to typical characteristic harmonics which are mainly a product of non linear loads.

Loads are the major source of characteristic harmonics and in some cases, in a very small degree, electric machines such as generators. Most harmonic pollution in a system is due to customer loads which have non-linear characteristics and demand distorted load currents from the system. These distorted currents behave as sources for harmonics, which can be considered harmonic frequency generators connected to the system. As was established by Fourier in his treatise, any non-sinusoidal periodic waveform can be fully described by a sum of multiples of the fundamental of the system frequency. Thus, a non-linear load having a waveform different than that of a perfect sine will be viewed through the system as a harmonic source. Harmonics produced by the "source" will be the components of the non-linear demand load. This creates a scenario where any non-linear load has become a sink for fundamental power and a generator for harmonic power. Consequently, any adjacent load will be receiving a distorted waveform produced by the superposition of the harmonic waveforms generated by the polluting load and the unpolluted power supplied by the utility. Ironically, loads may produce harmonics and be affected by them in return. Major harmonic offenders are power electronic devices which have non-linear characteristics and high power consumption.

Other important harmonic sources are arc furnaces used for industrial smelting and arc soldering equipment. Also Adjustable Speed Drives (ASDs) can cause severe voltage distortions due to high current draw and power electronics interfaces. Nonetheless, these are not the only loads that can be sources of harmonics in a power system, but they are representative of major offenders.

Since harmonics produced by a load can affect adjacent loads, the most effective way to treat harmonics is by their isolation and management at the load, preferably even within the load itself. Thus harmonics can be effectively controlled by using isolation devices such as transformers, UPS or by installing active or passive harmonic filters.

The other major power quality phenomena considered in detail are sags and swells. Sags are sudden voltage drops of one cycle to ten cycles of duration with of 10% or more in magnitude without service interruption. Swells on the other hand are voltage increases of 10% or more in magnitude for the same duration of time. Sags and swells are related to spikes and notches in several ways. Spikes and notches are also sudden voltage increases and decreases respectively, but have some critical differences. While sags and swells can have durations of multiple cycles, spikes and notches have durations of less than a cycle and are considered transient phenomena. Sags and swells are normally caused by the starting and shutdown of large electrical loads or local faults that do not cause system disconnection. Likewise, the sudden disconnection of certain devices will cause sudden increase in voltage levels, which constitute a swell. Spikes and notches are difficult to detect since their duration is very short, but they can be as damaging as sags and swells, and some times much more severe. Spikes are many times the result of indirect or close-by lightning strikes. The lightning strike will cause distortions in the magnetic field in the adjacent area strong enough to induce currents and over voltages in electrical circuits. Thus once a lightning strike occurs close to a power line, a spike can be created as a consequence of the energy discharge from the lightning. Since the energy contained in a lightning strike is concentrated in a short duration of time, spikes can be of great magnitude, and can cause damage in sensitive circuits and devices. Capacitor bank switching operations are also significant producers of spikes, including banks used for reactive injection and filter applications. Generally, sags and swells can cause equipment

failure due to un-served energy while spikes and notches cause damage by over-loading or over-stressing sensitive equipment.

Power quality problems are diverse and varied, and they cannot all be explained here in detail, but references [9] and [12] are definitive sources on the subject matter. It is clear that power quality is important to guarantee reliable and continuous service to customers. The issue that needs to be taken care of is how can power quality problems be efficiently studied and solutions be considered without having to experience the problems in the first place.

As with most engineering applications, many power quality methods are based on a battery of indices. Some of the most common ones are listed in Table 2.2.1.

**Table 2.2.1. Power Quality Indices**

| Index | Definition | (Eq.#) |
|---|---|---|
| *Total Harmonic Distortion (THD)* | $\frac{\sqrt{\sum_{i=2}^{\infty} I_i^{\,2}}}{I_1}$ | (2-1) |
| *Power Factor* | $\frac{P_{tot}}{\lvert V_{RMS}\rVert I_{RMS}\rvert}$ | (2-2) |
| *Telephone Influence Factor* | $\frac{\sqrt{\sum_{i=2}^{\infty} w_i^2 I_i}}{I_{RMS}}$ | (2-3) |
| *C Message Index* | $\frac{\sqrt{\sum_{i=2}^{\infty} c_i^2 I_i}}{I_{RMS}}$ | (2-4) |
| *IT Product* | $\sqrt{\sum_{i=1}^{\infty} w_i^2 I_i^2}$ | (2-5) |
| *VT Product* | $\sqrt{\sum_{i=1}^{\infty} w_i^2 V_i^2}$ | (2-6) |
| *K factor* | $\frac{\sum_{h=1}^{\infty} h^2 I_h^2}{\sum_{h=1}^{\infty} I_h^2}$ | (2-7) |
| *Crest Factor* | $V_{peak} / V_{RMS}$ | (2-8) |

| | | |
|---|---|---|
| *Unbalance Factor* | $\lvert V_- \rvert / \lvert V_+ \rvert$ | (2-9) |
| *Flicker Factor* | $\Delta V / \lvert V \rvert$ | (2-10) |

These indices are some of the most widely used for power quality applications [13]. Their common applications are presented next:

- ***THD:*** This is considered one of the most generally used indices for quality. Many utilities have actually adopted a THD based measure for their systems. It is mainly used to describe current distortion (harmonically induced) in systems and the devices which cause them [14][15].
- ***Power Factor****:* One of the more classical power system measurements; the true power factor (TPF or PF) as defined in (2-2), instead of the more classical, displacement power factor (DPF), is used for power quality measures. This is also applied similar to the THD, to determine customer distortion. [16][17].
- ***K-factor:*** Used to derate transformer ratings for high harmonic distortion applications. [18]-[20].
- ***Flicker Factor****:* This index has been used for service modulation problems. Recently, applications have been found with arc-furnace impact measurement.
- ***TIF:*** Telephone Influence Factor (TIF), is similar to the THD but applies weights to specific frequencies, with higher weights assigned to less desirable frequencies. Used to determine audio circuit interference.
- ***C-Message****:* Another circuit interference index, it also uses weights for calculation. The TIF is typically applied to analog communication lines, while the C-message is applied to digital circuits.
- ***IT and VT Products****:* Related indices, for current and voltage, respectively. Again, used for distortion measurements. Additionally IT product is used to determine stress on shunt capacitor banks [21].
- ***Crest Factor****:* This index is used to measure dielectric stress in materials.
- ***Unbalance Factor****:* Measures three-phase circuit unbalance.

As with all indices, their use can become inadequate if their applications and limitations are not considered. A power quality index, as all indices, tries to condense a complex

phenomenon into a number or set of numbers. Naturally, this will result in loss information on some fronts, which if ignored, can lead to serious oversights. Ambiguity in definition, index limitations and wrong application can foil the application of a power quality index. Ambiguity can affect in the following ways:

- ***Flicker Factor**:* This index measures the low frequency modulation of AC signals, which cause visual discomfort. It is defined for both an AC sinusoidal signal and a sinusoidal low frequency (lower than system frequency). The index does not consider non-sinusoidal modulation functions, and in many applications (if not most) the modulation will not be sinusoidal. Thus the calculation will enter into error if measurements are made considering it a periodic sinusoidal with RMS values.
- ***THD**:* The calculation of this index is defined only if a fundamental, system frequency, signal is present. If none is present, the THD becomes infinite. THD has also no weighting, so frequencies which are more harmful than others to power systems, are considered the same as those with little or no consequence.
- ***Power Factor**:* Power factor as defined by IEEE100-1992 [22] is given by equation (2-2), this is known in some circles as the *True Power Factor* (TPF). The classical definition of the power factor, $\cos(\theta)$ is a special case of this, and is known as the *displacement power factor* (DPF). This alone can cause confusion and mis-application. Yet, the use of the DPF in non-sinusoidal conditions is not defined, since it is a special case when no harmonic (or other) distortion is present. Furthermore, power factor itself is only defined for periodic signals.

## 2.3 Distribution Reliability / Performance Indices

Distribution reliability indices are widely used throughout the world to quantify the performance of electric service from utilities or specific utility areas. As the world adjusts itself to a liberalized-energy market, tools to quantify and qualify electric service have been sought by utilities, customers and regulatory agencies. To provide for a common method for calculating reliability indices, the IEEE produced in 1998 a draft standard [23] with recommended practices on calculation methods. This draft standard

evolved into the IEEE Standard 1366, *"Guide for Electric Power Distribution Reliability Indices,"* of 2001. The standard has been recently upgraded in 2004 to include a better definition of major events, and more advanced methods for determining major event threshold limits [1]. This standard provides methods for the calculation of reliability indices, which are in use throughout the world. As these indices are being adopted by utilities and regulators alike, they are being used to compare the performance of different utilities [1][3][23][24]. Also, utilities use the indices as a method to substantiate claims for infrastructure investment or operational improvement in their service areas [25]-[29].

The IEEE 1366-2003 standard provides indices for many applications. These include customer based and load based indices. Additionally, indices for both sustained and momentary interruptions are included in the standard. The indices included in the standard are the following:

- **Sustained Interruption Indices**
  - SAIFI – System average interruption frequency index
  - SAIDI – System average interruption duration index
  - CAIDI – Customer average interruption duration index
  - CTAIDI – Customer total average interruption duration index
  - CAIFI – Customer average interruption frequency index
  - ASAI – Average service availability index
  - ASAFI – Average system interruption frequency index
  - ASIDI – Average system interruption duration index
  - $CEMI_n$ – Customers experiencing multiple interruptions
- **Momentary Interruptions**
  - MAIFI – momentary average interruption frequency index
  - $MAIFI_E$ – momentary average interruption event frequency index
  - $CEMSMI_n$ – Customers experiencing multiple interruptions and momentary interruption events
- **Load Based Interruption Indices**
  - ASIFI – Average System Interruption Frequency Index
  - ASIDI – Average System Interruption Duration Index

### 2.3.1 Reliability Index Tracking

Outages to service have been shown, in the greater part, to be due to distribution events – up to 92% of total outages [3]. Thus, it is only natural to want to keep track of distribution network reliability and have efficient methods of doing so. Over the years there have been countless indices for this purpose, and as early as the 1970s, there were proposals for the use of indices for reliability measurement. Today, there are as many as 40 different reliability indices in use, and many more in internal use by utilities [3]. For distribution, the most useful measurement indices according to [30][31] are:

- Load interruption frequency (number/unit time)
- Expected duration of load interruption events (time)
- Total expected (average) interruption time per year (or reporting period)
- System availability or unavailability at the load supply point
- Unserved energy per year

Across the US, the reliability indices most often used are SAIFI and SAIDI. Closely related to these (and derived from), CAIDI, is another widely used index. SAIFI accounts for 83.33% while SAIDI 87.88% of usage across the board; CAIDI - 81.82% [32][33]. Table 2.3.1 shows some additional usage values for some indices.

**Table 2.3.1. Reliability Index Use**

| Index | Use |
|---|---|
| *SAIFI* | *83.33%* |
| *SAIDI* | *87.88%* |
| *CAIDI* | *81.82%* |
| *ASAI* | *66.67%* |
| *ASIFI* | *4.55%* |
| *ASIDI* | *7.58%* |
| *MAIFI* | *24.24%* |
| *CTAIDI* | *3.03* |
| *CAIFI* | *4.55%* |
| *Other* | *10.61%* |

*Source: Reference [33]*

With the initial proposal in 1998 of the IEEE1366 indices, Illinois was the first state to require reliability reporting that same year. After this milestone, over half of the

states have instituted a similar requirement or are considering it [3]. Even though momentary interruptions are not reported by many utilities (less than 25%), their impact can be considerable, as presented in reference [34].

The recent trend in regulation is to establish performance based rates (PBRs), where a utility is rewarded or penalized for their periodic performance. Hence, there is a need for a certain "standard" or baseline for comparison [3][24]. This has proven difficult to obtain, due to data variability. Difficulties arise on several fronts, including:

- Definition of outage for utility and regulators
- Indices to be used
- Reliability goals
- Definition of "poor" reliability
- Impact of storms
- Indices and their accuracy

The aforementioned points are areas of disagreement or discussion among different groups in industry, yet the trend is clear. There is a need for measurement of reliability, and more players are interested in it every day. Table 2.3.2 present data from reference [35]; it provides SAIFI/SAIDI data for different North American (including Canada) utility surveys. SAIDI values in hours for several utilities within a same state are presented in Table 2.3.3.

**Table 2.3.2. SAIDI/SAIFI Values for Several Surveys**

| | **SAIFI(INT/YR)** | | | **SAIDI(HR/YR)** | | |
|---|---|---|---|---|---|---|
| | **25%** | **50%** | **75%** | **25%** | **50%** | **75%** |
| ***IEEE 1366-1998]*** | *0.90* | *1.10* | *1.45* | *0.89* | *1.50* | *2.30* |
| ***EEI, 1999 Excludes Storms*** | *0.92* | *1.32* | *1.71* | *1.16* | *1.74* | *2.23* |
| ***EEI, 1999 Includes Storms*** | *1.11* | *1.33* | *2.15* | *1.36* | *3.00* | *4.38* |
| ***[CEA, 2001] Includes Storms*** | *1.03* | *1.95* | *3.16* | *0.73* | *2.26* | *3.28* |
| ***[PA Consulting 2001] Includes Storms*** | | | | *1.55* | *3.05* | *8.35* |
| ***Large City Survey [IP&L 2000]*** | *0.72* | *0.95* | *1.15* | *1.02* | *1.64* | *2.41* |

*Source: Reference [35]*

**Table 2.3.3. SAIDI Values for Several Utilities**

| **SAIDI for Several Utilities Within Same State (hr) (2003)** | | |
|---|---|---|
| *5.75* | *2.55* | *1.80* |

*Source: Reference [36]*

A newer idea behind reliability metrics is the theory of the *nines*. This metric of power availability is based on the percent of time that service is available to a customer. Modern utilities can today offer in the range of 99.0% to 99.9999% availability of service (two nines to six nines) with averages from 99.9% to 99.99% (3 to 4 nines). Typical rural and urban customers can expect levels of 2 to 3 nines and 5 to 6 nines, respectively. Table 2.3.4 gives the amount of time in minutes that service should be available for a given number of nines.

**Table 2.3.4. The Service Reliability by Number of Nines**

| The Standard of Nines | The Number of Nines | Minutes Off Supply |
|---|---|---|
| *0.9* | *1* | *52560* |
| *0.99* | *2* | *5256* |
| *0.999* | *3* | *525.6* |
| *0.9999* | *4* | *52.56* |
| *0.99999* | *5* | *5.256* |
| *0.999999* | *6* | *0.5256* |
| *0.9999999* | *7* | *0.05256* |
| *0.99999999* | *8* | *0.005256* |
| *0.999999999* | *9* | *0.0005256* |

An EPRI study in 2003 [3], found that the median SAIDI value across many US utilities (using ten years of data), was 107 minutes per year with a SAIFI of 1.1 interruptions per year. This data is comparable to results from other previous surveys [37][38].

# CHAPTER 3 ELECTRIC POWER DISTRIBUTION RELIABILITY INDICES

## *3.1* ***Statistics Theory***

All studies on statistics begin by the definition of the *sample space*. This sample space, typically denoted by letter *S*, is composed of all the possible outcomes of the experiment. Sample spaces are classified according to the number of elements they contain. They can be finite (discrete) or infinite (continuous) [39][40].

### 3.1.1 Probability Distributions

Probability distributions are functions which associate a value of a random variable to the probability of occurrence of this value [39][40]. Many have been developed, and are used to study innumerable natural and man-made phenomena. Among these, there are discrete value distributions and continuous value distributions. A discrete function can only take the form a set of predefined values, while a continuous function may take on a continuous range of infinite values.

Distributions have three main characteristics: skewness ($\gamma$), variance ($\sigma^2$) and mean ($\mu$). The skewness measures the amount of symmetry (or asymmetry) of a given distribution. When a distribution has a longer tail less than the maximum, the function has negative skewness, otherwise it exhibits positive skewness. Examples of skewness for varying values of standard deviation are presented in Figure 3.1.1.

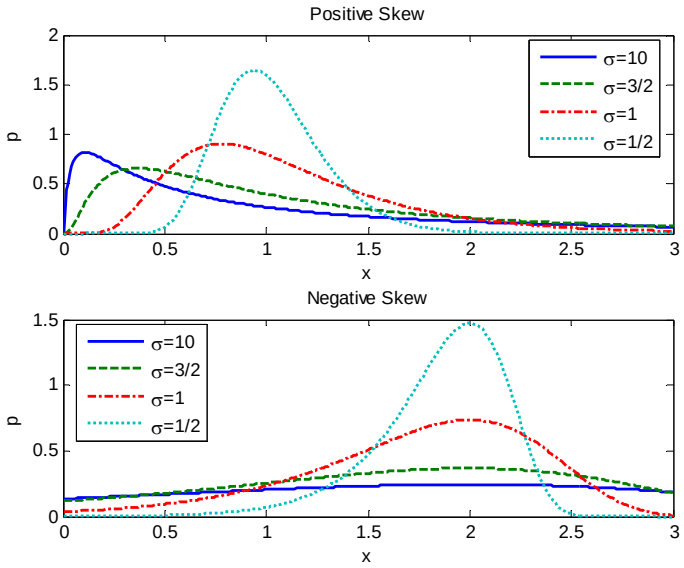



**Figure 3.1.1. Distribution Skew Examples**

The skewness of a distribution is defined to be:

$$\gamma_1 = \frac{\mu_3}{\mu_2^{3/2}}, \tag{3-1}$$

where $\mu_i$ is the *i*th central moment. The second moment about the mean is also know as the *variance* of the function. More detailed descriptions of the uses of moments and their deviations can be found in references [39][40][41]. Further information on distribution moments is beyond the scope of this work.

The *mean* of a sample space measures the center of the probability distribution in the sense of a center of gravity. The mean is given by,

$$\mu = \sum x \cdot f(x), \tag{3-2}$$

The relation of the mean to the moments of the function can be seen more clearly with Eq. (3-2). The first moment, or first derivative, of *f(x)* is the mean of the function. By definition, this is also the *first moment about the origin* of a discrete system of masses *f(x)* – the center of gravity of a physical object. The equation is not divided by $\sum_{allx} f(x)$, as

is usually done for the x-coordinate of the center of gravity, since this sum is by definition equal to 1. Higher moments about the mean are used to determine the skewness of the function, as stated in Eq. (3-2).

The *variance* of a function measures the dispersion of values from a central point, the *mean*. This can be achieved by obtaining the second moment about the mean of the function. It is given by,

$$\sigma^2 = \sum_{allx} (x-\mu)^2 \cdot f(x) \tag{3-2}$$

Unfortunately, this would not be in the same units or dimension as those from the values of the sample variable, so a more useful number is used, its square root. The square root of this value is know as the *standard deviation*,

$$\sigma = \sqrt{\sum_{allx} (x-\mu)^2 \cdot f(x)} \tag{3-3}$$

This value is now given in the units of the mean. When the sample standard deviation is divided by the sample mean it is known as the *normalized standard deviation*.

A *probability density function* (PDF), or probability function *P(x)*, is the derivative of a *cumulative density function* (CDF). Cumulative density functions are also known simply as density functions *D(x)*. CDFs describe the probability that a random variable *X* takes on a value less than or equal to a number *x*. The relation between continuous PDFs and CDFs is the following:

$$D(x) = P(X \le x) \equiv \int_{x_{\min}}^{x} P(x')dx' \tag{3-4}$$

If the functions are discrete, the relation is

$$D(x) = P(X \le x) = \sum_{X \le x} P(x) \tag{3-5}$$

Two statistical continuous distribution functions were used in this thesis. These were the *normal* and *log-normal* distributions, their descriptions and measurements are presented in latter sections.

#### *3.1.1.1 The Normal Distribution*

A normal distribution with random variable $X$ with mean $\mu$ and variance $\sigma^2$ is a statistic distribution with the probability function,

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)} \tag{3-6}$$

in the domain $x \in (-\infty, \infty)$. Its density function is,

$$D(x) = \frac{1}{2}\left(1 + erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right) \tag{3-7}$$

where *erf* is the Error Function given by:

$$erf(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt \tag{3-8}$$

This distribution is also known as the Gaussian distribution or the "Bell curve" due to its shape. The normal distribution was developed as an approximation to the binomial distribution. It has been used throughout history by Laplace in 1783 to study measurement errors and by Gauss in 1809 for the analysis of astronomical data [42]. In the normal distribution, the mean is given by $\mu$ while variance is equal to $\sigma^2$.

An application of this distribution is the description of outcomes for behavioral science test such as psychological and IQ tests. It is also helpful in the estimation of measurement error. Growth of living organisms and life expectancy of electric filaments can also find application for the normal distribution, among many others.

The standard normal distribution is a special case of the general normal distribution and is given by taking $\mu=0$ and $\sigma^2=1$ and changing variables to $Z=(X-\mu)/\sigma$, so that $dz=dx/\sigma$. Thus,

$$P(x)dx = \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz \tag{3-9}$$

Figure 3.1.2 present both the probability density function and the cumulative density function for a normal distribution.

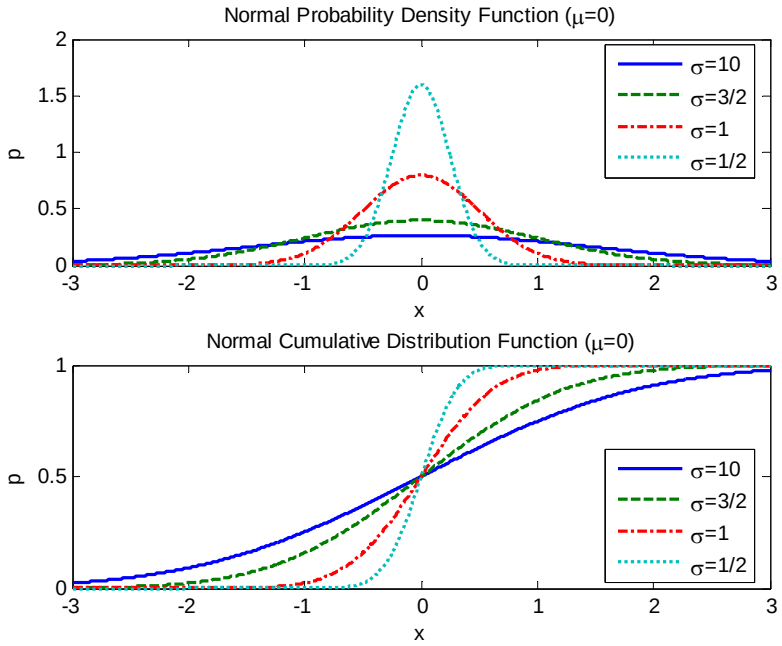



**Figure 3.1.2. Normal Distribution PDF & CDF**

### *3.1.1.2 The Log-Normal Distribution*

The log-normal distribution is a continuous distribution in which the logarithm of the random variable has a normal distribution. It is the general case of Gibrat's

distribution. A normal distribution occurs if the variable is the *sum* of a large number of independent, identically-distributed variables. On the other hand, a log-normal distribution occurs when the variable is a *product* of a large number of independent, identically distributed variables. The probability density and cumulative distribution functions for the log-normal distribution are:

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2\mu^2x^2}} e^{-(x-\mu)^2/(2\sigma^2)} \tag{3-10}$$

$$D(x) = \frac{1}{2}\left[1 + erf\left(\frac{\ln x - \mu}{\sigma\sqrt{2}}\right)\right] \tag{3-11}$$

The mean and variance are given by:

Mean: $E(x) = e^{(\mu+\sigma^2/2)}$ (3-10)

Variance: $V(x) = e^{2\mu}(e^{2\sigma^2} - e^{\sigma^2})$ (3-11)

This distribution is used in the study of particle distribution in natural aggregates and economic units, critical dosages in drug applications and the duration of doctor consultations among others [40][43]. It is very useful in the estimation of life expectancy for products that have a degrading performance, and failure times for fatigue growth in metals [41]. Figure 3.1.3 presents probability density and cumulative density functions for lognormal distributions.

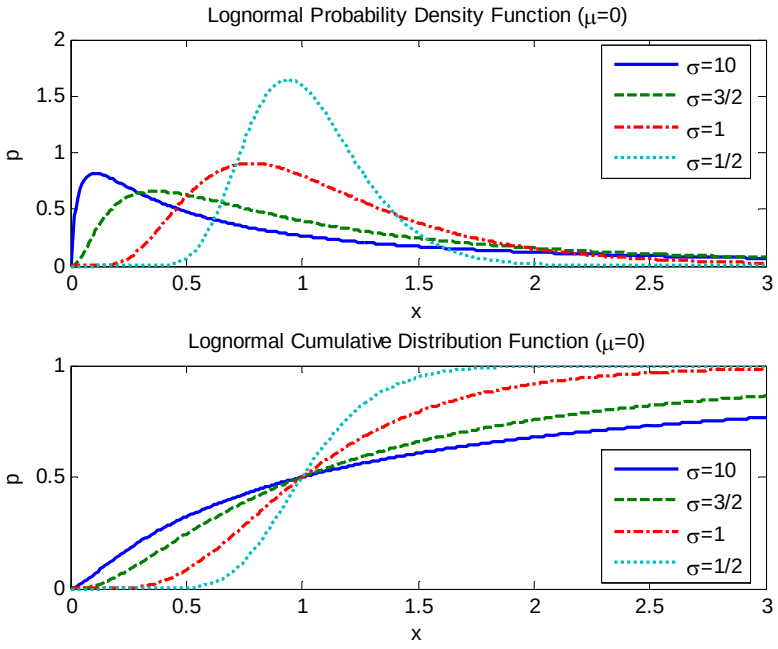



**Figure 3.1.3. Lognormal Distribution PDF & CDF**

## 3.1.2 Confidence Intervals

An important measurement when making statistical analysis is the *confidence interval*: an interval in which a measurement falls corresponding to a given probability. Usually the confidence interval is symmetrically placed around the mean. Generally this means that for a symmetrical PDF, the interval is evenly split in two halves around the mean. With the normal distribution, the probability that a measurement falls within $n$ standard deviations ($n\sigma$) of the mean $\mu$, within the interval [$\mu$-$n\sigma$, $\mu$+$n\sigma$], is given by:

$$P(\mu - n\sigma < x < \mu + n\sigma) = \frac{2}{\sigma\sqrt{2\pi}} \int_{\mu}^{\mu+n\sigma} e^{-(x-\mu)^2/(2\sigma^2)} dx \,, \tag{3-14}$$

After some manipulations and variable substitutions,

$$P(\mu - n\sigma < x < \mu + n\sigma) = \frac{2}{\sqrt{\pi}} \int_{\mu}^{\mu+n\sigma} e^{-\mu^2} du = erf\left(\frac{n}{\sqrt{2}}\right), \tag{3-15}$$

where *erf* is the so called *error function.* Table 3.1.1 summarizes the value of $P(\mu-n\sigma<x<\mu+n\sigma)$ that measurements from a normal distribution fall within $[\mu- x_n, \mu+ x_n]$ for $x_n=n\sigma$ with small values of $n$.

**Table 3.1.1. Probability for Typical Measurements**

| $x_n$ | $P(\mu - x_n < x < \mu + x_n)$ |
|---|---|
| $\sigma$ | *0.6826895* |
| $2\sigma$ | *0.9544997* |
| $3\sigma$ | *0.9973002* |
| $4\sigma$ | *0.9999366* |
| $5\sigma$ | *0.9999994* |

To find the probability confidence interval centered about the mean for a normal distribution in terms of σ, (3-15) has to be solved for $n$, thus

$$n = \sqrt{2} erf^{-1}(P) \tag{3-16}$$

where $erf^{-1}$ is the *inverse error function*. Table 3.1.2 gives the values of $x_p$ such that $[\mu- x_p, \mu+ x_p]$ is the probability ($P$) confidence interval for several widely used value of $P$.

**Table 3.1.2. Confidence Interval for Selected Probabilities**

| $P$ | $x_p$ |
|---|---|
| *0.800* | $1.28155\sigma$ |
| *0.900* | $1.64485\ \sigma$ |
| *0.950* | $1.95996\ \sigma$ |
| *0.990* | $2.57583\ \sigma$ |
| *0.995* | $2.80703\ \sigma$ |
| *0.999* | $3.29053\ \sigma$ |

Confidence intervals for log-normal distributions have a slightly different process, due to the skewness of the probability function. Nevertheless, it is a simpler process in some respects. In the log-normal distribution, the geometric mean and the geometric standard deviation are related. In this distribution, the geometric mean is equal to $e^{\mu}$ and the geometric deviation is equal to $e^{\sigma}$. In the normal distribution, confidence intervals are

determined using the arithmetic mean and standard deviation. Similarly, in a log-normally distributed population, these are obtained using the geometric mean and the geometric standard deviation. Lognormal confidence bounds are given in Table 3.1.3.

**Table 3.1.3. Lognormal Confidence Bounds**

| *Confidence Interval Bounds* | *Log space* | *Geometric* |
|---|---|---|
| *3 σ lower bound* | $\mu - 3\sigma$ | $\mu_{geo} / \sigma_{geo}^3$ |
| *2 σ lower bound* | $\mu - 2\sigma$ | $\mu_{geo} / \sigma_{geo}^2$ |
| *1 σ lower bound* | $\mu - \sigma$ | $\mu_{geo} / \sigma_{geo}$ |
| *1 σ upper bound* | $\mu + \sigma$ | $\mu_{geo}\sigma_{geo}$ |
| *2 σ upper bound* | $\mu + 2\sigma$ | $\mu_{geo}\sigma_{geo}^2$ |
| *3 σ upper bound* | $\mu + 3\sigma$ | $\mu_{geo}\sigma_{geo}^3$ |
| *Note:* $\mu_{geo} = \exp(\mu)$ *and* $\sigma_{geo} = \exp(\sigma)$ | | |

## 3.2 IEEE 1366 Distribution Reliability Indices

As mentioned in section 2.3, the IEEE has prepared a standardized method to calculate indices that quantify reliability for distribution networks. These indices rate networks according to their performance during a certain time period. The objects evaluated can be single feeders, areas, regions or entire systems. Generally, these indices are calculated in a monthly or yearly basis. With the latest installment of this standard, IEEE1366-2003 [1], improved definitions and methods have been implemented since the previous versions of 2001 and 1998 [23]. Primarily, changes to the standard have been based on the need for a more equitable method for determining Major Event Days (MEDs) for which all utilities, large or small, can benefit. The following sections describe the indices included in the standard and discuss the 2003 version of the MED determination process.

The indices are based on two general interruption categories: sustained and momentary interruptions. Within these two categories, indices that quantify the duration of the interruption and frequency of interruption are also included. The IEEE 1366

Standard provides an extensive list of indices based on customer or load data for quantifying reliability in a system. The indices included in the standard are the following:

- **Sustained Interruption Indices**
  - SAIFI – System average interruption frequency index
  - SAIDI – System average interruption duration index
  - CAIDI – Customer average interruption duration index
  - CTAIDI – Customer total average interruption duration index
  - CAIFI – Customer average interruption frequency index
  - ASAI – Average service availability index
  - ASAFI – Average system interruption frequency index
  - ASIDI – Average system interruption duration index
  - $CEMI_n$ – Customers experiencing multiple interruptions
- **Momentary Interruptions**
  - MAIFI – momentary average interruption frequency index
  - $MAIFI_E$ – momentary average interruption event frequency index
  - $CEMSMI_n$ – Customers experiencing multiple interruptions and momentary interruption events
- **Load Based Interruption Indices**
  - ASIFI – Average System Interruption Frequency Index
  - ASIDI – Average System Interruption Duration Index

Clearly, IEEE 1366 provides standard calculation methods for virtually every application. Nevertheless, four main indices are used throughout the world by utilities and regulators. Namely, the most used indices are: SAIDI, SAIFI, CAIDI, MAIFI and ASAI [3]. All indices are normalized by the amount of customers that are affected by an event, amount of events a single customer experiences or total load. The purpose behind this is to provide a way to compare results from different feeders or systems in which the customer load is not equal. Variable definitions for the equations are given in Table 3.2.1.

**Table 3.2.1. Factors for calculation of IEEE1366 Indices**

| ***Factors for calculating the Indices*** | |
|---|---|
| $r_i$ | *Restoration time for each interruption event* |
| $CI$ | *Customers interrupted* |
| $CMI$ | *Customer minutes interrupted* |
| $E$ | *Events* |
| $T$ | *Total* |
| $IM_i$ | *Number of momentary interruptions* |
| $IM_E$ | *Number of momentary interruption events* |
| $N_i$ | *Number of interrupted customers for each sustained interruption event during the reporting period* |
| $N_{mi}$ | *Number of interrupted customers for each momentary interruption event during the reporting period* |
| $N_T$ | *Total number of customers served for the areas* |
| $L_i$ | *Connected kVA load interrupted for each interruption event* |
| $L_T$ | *Total connected kVA load served* |
| $CN$ | *Total number of customers who have experienced a sustained interruption during the reporting period* |
| $CNT_{(k>n)}$ | *Total number of customers who have experienced more than n sustained interruptions and momentary interruption events during the reporting period* |
| $k$ | *Number of interruptions experienced by an individual customer in the reporting period* |
| $T_{MED}$ | *Major event day identification threshold value* |

## 3.2.1 Sustained Interruption Indices

The most used indices are SAIDI and SAIFI, and both complement each other. SAIDI is the system average interruption duration index. In other words, the value provided by this index gives an idea of the average duration of total service outage that is experienced by clients in a system. Conversely, SAIFI provides an idea of the amount of interruptions a certain client in a system experiences in the analysis timeframe.

### *3.2.1.1 Sustained Average Interruption Frequency Index (SAIFI)*

This index in conjunction to SAIDI, are the most widely used indices to quantify service performance [3][4]. SAIFI quantifies the amount of interruptions the average customer will experience during the course of the reporting period. Let it not be understood that all customers will experience the same amount of events; unlucky end-of-

feeder clients will experience more interruptions than those upstream on the feeder. Numbers provided by this index are average. SAIFI is also known as the Average Failure Rate, labeled as λ in most reliability publications such as IEEE493-1997[IEEE493]. Another well used value, the Mean Time Before Failure (MTBF) is equal to 1/λ or 1/SAIFI. SAIFI is obtained with the following,

$$SAIFI = \frac{Total \quad number \quad of \quad customer \quad Interruptions}{Total\, number \quad of \quad customers \quad served} \tag{3-17}$$

$$SAIFI = \frac{\sum N_i}{N_T} \tag{3-18}$$

#### *3.2.1.2 Sustained Average Interruption Duration Index (SAIDI)*

Similar in application to SAIFI, this index provides average total interruption values for the customer base in the reporting period. It provides an average value which estimates the amount of hours or minutes the customer was without service. Other names used in literature for this index are CMI and CMO (Customer Minutes of Interruption or Outage, respectively). SAIDI is obtained with the following,

$$SAIDI = \frac{\sum Customer \quad Interruption \quad duration}{Total \quad number \quad of \quad customers \quad served} \tag{3-19}$$

$$SAIDI = \frac{\sum r_i N_i}{N_T} \tag{3-20}$$

#### *3.2.1.3 Customer Average Interruption Duration Index (CAIDI)*

Related to both SAIDI and SAIFI, CAIDI is another index used for quantifying reliability. It provides an apparent time of repair averaged over the affected customer base. This number is quite generally an artificial number, since outages are attended by sectionalizing feeders which provide for some customers with little downtime, and others with considerable more repair time. CAIDI is obtained with the following,

$$CAIDI = \frac{\sum Customer \quad Interruption \quad Duration}{Total \quad Number \quad of \quad Customers \quad Interrupted} \tag{3-21}$$

$$CAIDI = \frac{\sum r_i N_i}{\sum N_i} = \frac{SAIDI}{SAIFI} \tag{3-22}$$

#### 3.2.1.4 Customer Total Average Interruption Duration Index (CTAIDI)

CTAIDI and CAIFI are related in the sense they are both hybrid indices. The values obtained for these indices are for the customers experiencing the loss of service. It is not averaged for all the customers in the area, or feeder. CTAIDI provides the total average time experienced in outage for those affected. CTAIDI is obtained with the following,

$$CTAIDI = \frac{\sum Customer \quad Interruption \quad Duration}{Total \quad Number \quad of \quad Customers \quad Interrupted} \tag{3-23}$$

$$CTAIDI = \frac{\sum r_i N_i}{CN} \tag{3-24}$$

It is important to point out that in tallying *Total Number of Customers Interrupted*, each individual customer should only be counted once regardless of times interrupted during the reporting period. This applies to both CTAIDI and CAIFI.

#### 3.2.1.5 Customer Average Interruption Frequency Index (CAIFI)

Very similar to CTAIDI, CAIFI provides the average frequency of sustained interruptions for those customers actually affected by the interruptions. The customer is counted once, regardless of the times interrupted in this calculation. CAIFI is obtained using the following expressions,

$$CAIFI = \frac{\sum Total \quad Number \quad of \quad Customers \quad Interrupted}{Total\ Number \quad of \quad Customers \quad Interrupted} \tag{3-25}$$

$$CAIFI = \frac{\sum N_i}{CN} \tag{3-26}$$

#### 3.2.1.6 Average Service Availability Index (ASAI)

This index provides the fraction of the time within a reporting period that the client has had service, as shown in Eq. (3-27). This index is used mainly for industrial applications [4]. ASAI is obtained with,

$$ASAI = \frac{Customer \quad Hours \quad Service \quad Availability}{Customer \quad Hours \quad Service \quad Demands} \tag{3-27}$$

$$ASAI = \frac{N_T \times (Number \quad of \quad Hours/yr) - \sum r_i N_i}{N_T \times (Number \quad of \quad Hours/yr)} \tag{3-28}$$

#### 3.2.1.7 Customers Experiencing Multiple Interruptions (CEMIn)

This is a special application index, used to derive the amount of customers experiencing a defined amount of interruptions versus total number of customers. $CEMI_n$ is obtained with the following expression,

$$CEMI_n = \frac{Total \quad Number \ of \quad Cust \quad that \quad Experience \ More \ than \ n \quad Sust \quad Inter.}{Total \quad Number \ of \quad Customers \ Served} \tag{3-29}$$

$$CEMI_n = \frac{CN_{(k>n)}}{N_T} \tag{3-30}$$

### 3.2.2 Load Based Interruption Indices

Load based indices like those described in sections 3.2.2.1 and 3.2.2.2, are difficult to use for residential applications, since power demand is not easily obtained nor recorded. These indices are better suited for industrial applications. Consequently they

find little application, since large industrial customers have enough influence to force utilities to solve problems without the need for indices.

#### *3.2.2.1 Average System Interruption Frequency Index (ASIFI)*

This index is very similar to SAIFI and homogenous in application, but is based on load instead of customers, thus suited for areas of high concentrations of load and few customers. In a system with homogenous load distribution, ASIFI would equal SAIFI. ASIFI is obtained with the following,

$$ASIFI = \frac{\sum Total \quad Connected \quad kVA \quad of \quad Load \quad Interrupted}{Total \quad Connected \quad kVA \quad Served} \tag{3-31}$$

$$ASIFI = \frac{\sum L_i}{L_T} \tag{3-32}$$

#### *3.2.2.2 Average System Interruption Duration Index (ASIDI)*

Homologous to SAIDI but based on load, faces the same limitations as those for ASIFI. ASIDI is obtained with the following,

$$ASIDI = \frac{\sum Connected \quad kVA \quad Duration \quad of \quad Load \quad Interrupted}{Total \quad Connected \quad kVA \quad Served} \tag{3-33}$$

$$ASIDI = \frac{\sum r_i L_i}{L_T} \tag{3-34}$$

### 3.2.3 Momentary Interruption Indices

Additional indices are used to describe the frequency and average total duration of momentary interruptions. These are presented below.

### 3.2.3.1 Momentary Average Interruption Frequency Index (MAIFI)

This index provides interruption frequency based on momentary interruptions. Similar to SAIFI, but using momentary events; sustained events are not considered. MAIFI is obtained with the following,

$$MAIFI = \frac{\sum Total \quad Number \quad of \quad Customer \quad Momentary \quad Interruptions}{Total \quad Number \quad of \quad Customers \quad Served} \tag{3-35}$$

$$MAIFI = \frac{\sum IM_i N_{mi}}{N_T} \tag{3-36}$$

### 3.2.3.2 Momentary Average Interruption Event Frequency Index (MAIFIE)

This index indicates average frequency of momentary interruptions, same as MAIFI, but unlike MAIFI it does not include the events immediately preceding a lockout. $MAIFI_E$ is obtained with the following,

$$MAIFI_E = \frac{\sum Total \quad Number \quad of \quad Customer \quad Momentary \quad Interruption \quad Events}{Total \quad Number \quad of \quad Customers \quad Served} \tag{3-37}$$

$$MAIFI_E = \frac{\sum IM_E N_{mi}}{N_T} \tag{3-38}$$

### 3.2.3.3 Customers Experiencing Multiple Sustained Interruption and Momentary Interruption Events (CEMSMIn)

This index is used for individual customers if average methods cannot be used to identify problems. It provides the ratio of individual customers experiencing more than a specified number of sustained and momentary interruptions to the total customers served. $CEMSMI_n$ is obtained with the following

$$CEMSMI_n = \frac{Total\ \ Num.\ \ of\ \ Cust.\ \ Experiencing\ \ Mora\ \ Than\ \ n\ \ Inter.}{Total\ \ Number\ \ of\ \ Customers\ \ Served} \quad (3\text{-}39)$$

$$CEMSMI_n = \frac{CNT_{(k>n)}}{N_T} \quad (3\text{-}40)$$

### 3.2.4 Major Event Days

A Major Event Day is a classification assigned to a day in which a certain occurrence prevents the proper or normal operation of a power system. This day is characterized by severe number of blackouts and/or long duration interruptions of service [44]-[48]. Since the severity of the events is relative, MED definitions have been a source of dispute since their application to reliability indices. As is natural, utilities want to declare more days as MEDs since it helps improve their reliability numbers, while regulators seek the best indicators for actual system reliability. Regulators are sometimes reluctant to exclude any events since any interruption, no matter its cause, is still energy unserved and a corresponding impact on clients.

With the original and subsequent versions of the IEEE1366 Standard, different and varying definitions of what constitutes a major event have been formulated by numerous utilities, regulators and operators. For example, some regulators define major events as those which cause failure of service to 10% of the total customer base, or 10% in a given operational area. Still others consider a major event any which causes interruptions of 24 consecutive hours or more. Some utilities even include maintenance operations as a major event, since it does not exemplify typical system operation. Thus major event definitions are all over the map [3][33][37].

To standardize the process of defining major events, the latest version of the IEEE1366 Standard, proposes a statistical method, the 2.5Beta. The idea behind establishing a mathematical basis for defining major events, other than preventing the use of a myriad of diverging definitions, is to provide utilities and regulators an efficient way to process everyday data [44]-[48]. The idea behind processing MEDs separately provides a way to study normal system operation more accurately. Once a day is declared a MED, its data is extracted and studied separately from common, non-MED days. This

allows the analysis to provide better day to day tracking and reveal trends in operation without the serious data point outliers that MEDs can inject into databases; minimizing the great impact on variance that a MED can cause. This way the variance present in the data is due to normal system operation and not a severe storm, fire or machine failure. This way the system analysis is more concentrated in system performance itself rather than how bad the events and their effects on the system were [3][46].

The underlying equation of the 2.5Beta is the following:

$$T_{MED} = e^{(\alpha + 2.5\beta)} \tag{3-41}$$

The $T_{MED}$ value is known as the *MED threshold value.* This value will determine if a certain day is a MED or not. The value is compared to the daily SAIDI value; if the SAIDI value is higher, then that day is declared a MED and it is separately studied and is not included for that reporting period. The process to determine the $T_{MED}$ value and MEDs is as follows:

1. Daily values for SAIDI are calculated and stored. Preferably, this should be five consecutive years. If five years are not available, collect the most years possible and complete the process, until five years previous to the year under study are obtained. No more than five years are recommended [1]
2. Use only daily SAIDI values which are different from zero.
3. Obtain the natural log (ln) of each SAIDI value.
4. Obtain alpha (α), the average of the log values.
5. Obtain beta (β), the standard deviation of the log values.
6. Calculate $T_{MED}$ using equation (3-41).
7. Days with SAIDI values higher than $T_{MED}$ are excluded from the subsequent reporting period.

## 3.3 *Factors Affecting Reliability*

According to IEEE100-1992, an interruption to service is *the isolation of an electrical load from the system supplying that load, resulting from an abnormality in that system* [21]. The abnormality in the system can either be a malfunction of a system component, a fault or a system operation due to any of the aforementioned. Interruptions, independent from the cause, are generally undesired, as they leave energy unserved and customers without service. Most of the time, interruptions occur because the system is reacting to a fault. A fault or short-circuit is defined by IEEE100-1992 as *an abnormal connection (including an arc) of relatively low impedance, whether made accidentally or intentionally, between two points of different potential* [22]. Thus, interruptions are mostly reactions of the protection system to unexpected fault conditions or operator-requested de-energizations. Interruptions are the reason for the study of system reliability, yet many factors come into play with reliability numbers. Variables affecting distribution reliability numbers are:

- ***Definitions and Classifications***
  - Major Events
  - Interruptions/Outages
  - Planned/Unplanned Events
  - Distribution/Transmission
- ***Data Collection and Accuracy***
  - Outage Notification
  - Outage Reporting
  - Restoration Process
  - Customers
- ***Service Territory***
  - Geography
  - Weather
  - Flora/Fauna
  - Vehicular Traffic
- ***System Topology and Design***

- Urban/Rural/Downtown
- Load Characteristics
- Load Density
- Underground/Overhead
- Voltage level
- Protection Scheme

Since interruptions and faults are such a critical subject of study, more review of these is warranted. Interruption information and discussions on factors affecting reliability are discussed in latter sections.

Note that not all factors affecting reliability numbers are due to the faults or interruptions themselves, but to human factors. Definitions, data collection and accuracy severely affect reliability results. The first factor to consider is the definition of what an interruption is. Some utilities, exclude planned outages and outages due to failures outside the distribution system from index calculation. This creates variability and not all results will be based on the same conditions and results are not directly comparable. Additionally, another detail that fogs indices is starting and ending times of an interruption. Some utilities use the SCADA data, others use dispatch crew logs, others use the customer service database using call times [29][37][38][49]-[51]. These are just examples of definitions used by utilities for establishing what an interruption is, and how they measure them.

On a survey completed shortly after the approval of the IEEE1366-1998 document, several utilities reported their definitions and practices for reliability calculations [38]. It was found that not only utilities clash on definition of interruption, but also on what is the equipment to include in the calculations. Table 3.3.1, presents the equipment considered by utilities responding to the survey.

**Table 3.3.1. Equipment included in calculations**

| **Equipment Included** | |
|---|---|
| *Generation* | 6% |
| *Transmission* | 12% |
| *Distribution Substation* | 14% |
| *Circuit Breaker* | 14% |

| | |
|---|---|
| *Recloser* | *12%* |
| *Sectionalizer* | *11%* |
| *Fuse* | *11%* |
| *Transformer Only* | *10%* |
| *Service* | *7%* |
| *Meter* | *5%* |

*Source: Reference [38]*

### 3.3.1 Interruption and Fault Types

Interruptions can occur for two main reasons, de-energization of a circuit for maintenance or repair or a fault occurs. Interruptions can be momentary or sustained. According to the IEEE1366, a momentary interruption is any lasting less than 5 minutes [1]. This would include any reclosing up until lockout. Once lockout occurs and the interruption lasts more than 5 minutes, it is considered a sustained interruption. Momentary interruptions are mostly caused by automatic reclosing, but in rare cases, they may be due to intermittent self-clearing faults [4].

Faults or short-circuits are the major cause of interruptions. These can be categorized in two main categories: temporary and permanent. Temporary faults account for the majority of faults in distribution systems, with most systems experiencing anywhere from 60 to 80% from total system faults. Nevertheless, permanent versus temporary fault numbers vary greatly from utility to utility, with some experiencing even splits between the two [4]. Temporary faults can occur for a myriad of reasons, but may include tree or animal contact and weather as the main contributors. Temporary faults are easily solved, with little or no intervention from the system itself. Many are self-clearing, such as a branch or animal contact which burn and fall off, conductors slapping together in severe wind or insulation flashover due to contamination. Lightning is also a temporary fault. Lightning arrester failure, on the other hand, can become a permanent fault. Other temporary faults are simply cleared once a trip from the substation is issued. Instantaneous reclosing de-energizes the line for a short duration of time, which allows the arc or contact path to disappear, which in turn eliminates the fault path. Once the circuit is re-energized, the system resumes normal operation. Permanent or persistent faults, on the other hand, are those that cannot be solved with reclosing action and will

not self-clear. This kind of fault requires attention from system operators and generally requires the use of line crews to repair. Equipment malfunction, cable failure, downed lines or persistent tree contact can all produce permanent faults. It is important to point out, that some tree contact can cause permanent faults, such as a tree falling on a line. Persistent tree contact can cause fuses in laterals to blow, which then requires a crew to replace the fuse(s). Typical faults causes are tabulated in Table 3.3.2.

**Table 3.3.2. Fault Causes by Type**

| Fault Causes | |
|---|---|
| **Temporary** | **Permanent** |
| *Lightning* | *Trees & Tree Limbs (Fallen)* |
| *Animals* | *Cable Failure* |
| *Trees (Wind/Intermittent)* | *Equipment Failure* |
| *Insulation Contamination* | *Trench Dig-in* |
| *Wind/Weather* | *Accident* |

## 3.3.2 Factors Affecting Reliability

Fault causes are varied and variable, but they tend to follow certain trends. Incidence varies greatly from area to area, and season to season, for faults can depend greatly on climate and local environment [49][50]. According to [4], tree contact and lightning are the major fault causes, amounting to approximately 20% each. Other causes are equipments failure, animal contact, wind, dig-ins and accidents. Table 3.3.3, provides fault cause data from three different sources, an IEEE study in 1983 [52], and references [37] and [50]. Note that numbers and categories vary considerably.

**Table 3.3.3. Fault Causes**

| Cause of Failure | [37] | [32] | [50] |
|---|---|---|---|
| *Accident/Vehicle Accident/Public Accident* | *3%* | *2%* | *10%* |
| *Age* | *11%* | | |
| *Animals* | *16%* | *22%* | *18%* |
| *Construction Activity* | | *>1%* | |
| *Customer Equipment* | *7%* | | |
| *Dig-in* | *6%* | *3%* | |
| *Equip Failure* | *5%* | *11%* | *14%* |
| *Ice/Snow* | | *>1%* | |
| *Lightning* | *16%* | *16%* | *9%* |
| *Other* | *11%* | *23%* | *13%* |
| *Overload* | *2%* | | |

| | | | |
|---|---|---|---|
| *Storm* | *13%* | | |
| *Tree Contact* | *10%* | *16%* | *19%* |
| *Unknown* | | | *17%* |
| *Vandalism* | | *>1%* | |
| *Wind* | | *5%* | |

Note: Missing data, indicates no such category.

Modern three-phase distribution systems can experience different forms of a short circuit. These can occur from phase to phase, any number of phases to neutral and any number of phases to ground (broken conductor). Of all the possible forms, the most typical is the single-phase to neutral. This is due to the fact any number of objects with a zero potential can come into contact with an energized line in one point easier than with two or three. The other most common fault is the phase-to-phase type. This kind of fault is mainly due to animal or tree contact, which is a very high occurrence event. Nonetheless, trees can also cause line-to-ground faults, especially under rainy or moist conditions. Multiple-phases to ground events are severely limited in occurrence, and are mainly the result of equipment failure. Faults involving broken conductors, in which a live line physically touches the ground, are also of the single-phase kind in most occurrences, for the same reasons. Table 3.3.4 gives some approximate numbers of fault occurrences.

**Table 3.3.4. Number of Phases Involved in Fault**

| Fault | Percentage |
|---|---|
| *Phase to phase* | *11%* |
| *One Phase* | *65%* |
| *Two phase* | *2%* |
| *Three phase* | *2%* |
| *One phase on the ground* | *15%* |
| *Two phases on the ground* | *1%* |
| *Three phases on the ground* | *>1%* |
| *Other* | *5%* |

*Source: Reference [52]*

Fault causes and fault types are used to create fault code libraries. Fault code libraries are used by utilities to efficiently log, archive and evaluate interruption events throughout their systems. These generally assign categories to several types of fault causes. For example, some common categories are: equipment failure, tree contact and animal interference. These categories simplify the analysis of fault occurrences by

eliminating the need for complex descriptions. Fault codes are used in anything from dispatching service crews to system performance and improvement studies.

Service territory and system design play a vital role on how reliability numbers play out. Even protection systems and their operation schemes take a toll on reliability numbers [4][28] Service territory will determine the geography and patterns (i.e. vegetation, vehicular, animal population, weather) of the area. These factors affect considerably fault causes and fault occurrence. System design is closely tied to this, since the networks topology, protection scheme, right-of-ways, voltage level among others will determine the system's response and exposure to the factors established by the service territory. Not to be underestimated, crew response and efficiency is affected by both factors (service territory and system design) and are instrumental to good reliability numbers. Table 3.3.5 presents some typical SAIFI and SAIDI ranges for different topologies [15].

**Table 3.3.5. Reliability of Network Topologies**

| | SAIFI (int/yr) | CAIDI (min/int) | MAIFI (momentary/yr) |
|---|---|---|---|
| *Radial* | *0.3 to 1.3* | *90* | *5 to 10* |
| *Primary auto-loop* | *0.4 to 0.7* | *65* | *10 to 15* |
| *Underground residential* | *0.4 to 0.7* | *62* | *4 to 8* |
| *Primary selective* | *0.1 to 0.5* | *180* | *4 to 8* |
| *Secondary selective* | *0.1 to 0.5* | *180* | *2 to 4* |
| *Spot Network* | *0.02 to 0.1* | *180* | *0 to 1* |
| *Grid Network* | *0.005 to 0.02* | *135* | *0* |

*Source: Reference [3]*

Service territory significantly affects SAIDI/SAIFI numbers. Table 3.3.6 presents SAIDI values for different service territories within Australia. The different categories are: Commercial Business District (CBD), essentially a downtown category, urban, rural and remote.

**Table 3.3.6. SAIDI Values for various Australian Service Territories**

| Network Type | Year | | | | |
|---|---|---|---|---|---|
| | 1995-1996 | 1996-1997 | 1997-1998 | 1998-1999 | 1999-2000 |
| CBD | | | *23.5* | *5.1* | *12.4* |
| Urban | | *83* | *85.8* | *93.1* | *99.5* |
| Rural | | *164* | *143.4* | *130.6* | *141.9* |

| | | | | | |
|---|---|---|---|---|---|
| Remote | | *300* | *244.2* | *230.6* | *237.0* |
| Total SAIDI | *116* | *118.4* | *112.6* | *110.4* | *118.6* |

*Source: Reference [3]*

For example, in a highly wooded area, one can come to the conclusion that tree faults are the bigger culprit for interruptions. But this may not be true in the city, where there not only are no trees (or very little, and purely ornamental) but most circuits are underground. Thus, for the city, 20% of tree induced faults cannot be true. This shows that one must carefully study fault percentages. Not only do the fault cause percentages differ, but also the exposure. An overhead line crossing a desert will be exposed to little or no weather induced faults, unlike one crossing a tropical rain forest. Each one will have its exposure problems, but highly different one from the other.

## *3.4 Comparative Framework*

It is the object of this study to provide a more adept method for determining distribution network reliability. It is not the intention of the author to create a new set of indices to measure or quantify reliability in distribution networks, but to adapt current indices and methods to better suit the myriad of system types. Thus a framework to adapt current IEEE1366 indices is planned. This framework would help to close the gap on the discrepancy between different systems or areas, and better comparison among systems or areas.

### 3.4.1 Objectives of framework

The framework has one main purpose, and that is to allow the adjustment of reliability index results to provide a leveled comparison base and allow the cross evaluation of results from totally distinct systems. In other words, the method should minimize the unique factors of each area, and provide results which are more coherent.

Reliability indices are universally used as performance indicators for utilities and their service, but they can be misleading in the information they provide. As with all indices used in the electric industry, they are mainly used to compare results from dissimilar specimens, in this case, distribution service reliability. Additionally,

distribution reliability's relationship with power quality indices will be studied and possible index interactions will be considered.

A typical problem with using any kind of indices is their deceptively simple application. The results provide a lot of insightful information, but can also miss a great amount of relevant characteristics. That is, indices can provide for results devoid of the original context of the measurements or data. Thus, even though results from different sources may appear to be in coherent form, it may not be a valid comparison due to context considerations. Disregarding the circumstances from which the results were obtained invalidates the findings, unless a common ground for comparison is reached. This is also the case with reliability indices, where topology, circuit types, geography, climate, among other factors seriously affects data. As a result, indices for a utility are a unique class in itself and not directly correlated with those of others.

To validate the use of reliability indices for comparison between different utilities, a common ground must be met. For this common basis to be established the values have to be adjusted in such a way that feeder and system distinctiveness is not arbitrarily stripped from the index results, but compensated or equalized. Only after this embedded context data is processed, can comparisons be made. This study has the purpose to develop the basis for this system of index compensation, to provide for one on one comparison capabilities. For the study, actual reliability data will be used.

### 3.4.2 Proposed Method

A great number of factors affect the performance and reliability of distribution feeders. These have been discussed in previous sections. Nevertheless, an accurate interruption database can be instrumental in the interpretation of system characteristics, without studying the system directly. If a utility consistently uses a fault code database for interruptions during reporting periods, it can be used to decipher system characteristics if these are not available or are not made available. Obviously having the system data, the interruption data and the fault code data is a perfect scenario, yet rarely plausible.

The proposed method is based on *the use of fault codes to complement and adjust reliability data according to region/area characteristics*. A region's or system's uniqueness is contained within the interruption/fault data. Thus this data itself can be used to create profiles for regions and to adjust or normalize results within different systems and/or areas.

For the project, the method proposed is based on the use of the fault code data in conjunction with the interruption data to obtain a better estimate of system performance. The method will work for any system or area as long as the number of fault codes is the same or equivalences can be made. The investigator additionally, sees no limitation in applying this method to different systems or different areas within a same system because the underlying principles are the same. The requirement for all areas is to have the same amount of fault codes or equivalency be made of them, or a certain number of them used.

The 2.5Beta process of the IEEE1366 uses the daily SAIDI value for the calculation of $T_{MED}$. The reason for this is that the SAIDI index is a good indicator of operational and design stress of the system. Interruptions will happen on any system, it is an inevitable fact, but how the system reacts to them is very dependent on the system design and operation. Thus, an index such as SAIFI which only measure interruptions is not used. For the same reasons, the SAIDI index will be used for the proposed method. The other main contender for this application would be ASIDI. Unfortunately, as mentioned in section 3.2.2, load data can be hard to come along, while customer data is relatively simple to get, and is easily estimated by crews if it is not available. Nevertheless, the principles behind the proposed framework could have been based on ASIDI or similar indices if data were available. Thus, the framework presents a flexible tool for distribution system analysis.

The proposed method does not plan to alter SAIFI values. SAIFI results will not be modified in any way different from the method from IEEE1366. The reason for this is that the method seeks to find a way to adjust performance of different systems, but it does not aim to artificially try to minimize interruptions. Thus the interruption frequency number should remain the same, even though the weights of some interruptions differ from others, which only would affect apparent duration.

The method will be as follows: An initial step produces SAIDI and SAIFI results for the system using strictly the IEEE1366 process. In the process, the fault codes are extracted from the database for the interruptions included in the SAIDI calculation. The fault codes are collected for each system under study. Fault codes to be used are selected and/or made equivalent. Profiles for each system are produced with the fault codes extracted. This is done by calculating the totals for each type of fault for each region. The profiles are used to determine characteristics of the system.

These values will be used to create a weight matrix for the adjustment that will come along. A weight matrix will be used for the adjustment for SAIDI/SAIFI. Several methods for the creation of weights will be evaluated. After weights are created, IEEE1366 processes are completed with the adjustments from the weights. The adjusted yearly value is then obtained as before. No change to the IEEE1366 process will be made, except for the multiplication of the weight value for each interruption during the process.

# CHAPTER 4 COMPARISON FRAMEWORK IMPLEMENTATION

The previous chapter gave an overview of the process to be completed for this project, yet not many details of its implementation. That is the purpose of this chapter, to discuss why and how the framework process was implemented. It includes a verification process of the data, which is described in section 4.1. The reliability calculations and all related steps are discussed in section 4.2. Following that, major event considerations are explained in section 4.3. Fault code data analysis is discussed in section 4.4. Finally, section 4.5 gives the weighting procedure detail and discussion.

## *4.1 Data Verification*

A verification step was completed before any calculations, to extract any bad data. Additionally, the statistical distribution of the original data was sought. A data screening process was implemented in Matlab to check the interruption data, and it was also inspected manually to pickup any hard to detect errors. Bad dates were removed or replaced if the repair was obvious. For example, if interruptions of several years duration were found, the year entry for both dates was verified. The year which was believed incorrect was replaced with the one closest to the previous event in the database. This same procedure was used for suspected bad month data. If a data entry was un-repairable, it was eliminated to prevent any contamination of the data. In our case, less than 100 samples had to be repaired or eliminated from sample space of approximately of 500,000 entries. Below, a sample code for the Matlab cleaning script.

**Sample Verification Code:**

```
for i=1:reg_0_size(1,1)

   if abs((reg_0(i,6)-reg_0(i,7)))>tolerance
      badrows(j,1)=i+1;
      badrows(j,2)=reg_0(i,6);
      badrows(j,3)=reg_0(i,7);
      badrows(j,4)=reg_0(i,8);
      badrows(j,5)=reg_0(i,13);
      badrows(j,6)=reg_0(i,11);

      j=j+1;
   end
end
```

Following the purging and repair of bad data, its statistical distribution was to be determined. According to several sources [1][4][53], power distribution interruption occurrence data follows a log-normal distribution. Logically, this should be true, since the skewness of the distribution would indicate that the system is operating correctly most of the time, while interruptions are occasional. This is true even for the worst system. If it would follow a normal distribution, for example, it would indicate that it operates half the time and the other half of the time it would be out of service; undesirable indeed. The log-normal distribution has several characteristics, namely:

- The median, not the average is a better measure of the average or typical customer. The average is higher than the median, due to the skewed distribution.
- Worst performing feeders dominate the indices, which are weighted average values.
- Outliers skew the indices; outliers are mainly, major events

If a dataset follows a log-normal distribution, the natural logarithm of its values conform to the normal distribution. The daily values for SAIDI were collected from the system and their logs were calculated. This new log-set was then plotted and compared to the normal line. This is very typical experiment to verify the normality of data and given in many statistic references [39]-[41]. Figure 4.1.1 illustrates one of the areas under study.

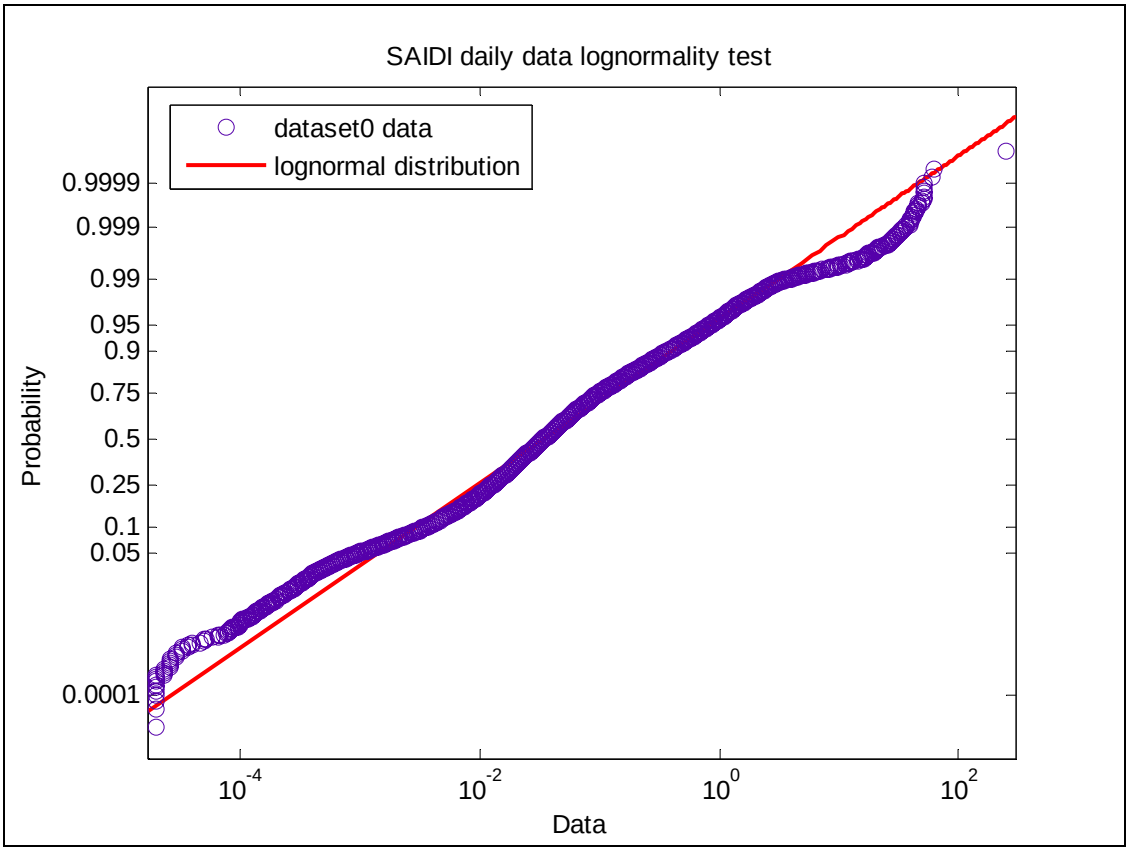



**Figure 4.1.1. Lognormality test results**

It is clear from Figure 4.1.1, that the curve from the log-set follows quite closely the normal line. Other areas under study exhibited very similar results and it was deemed unnecessary to present here, but are included in the appendices. As illustrated in Figure 4.1.1, the system under consideration conforms to a lognormal incidence of events as expected. Note that at the extremes of the line, the data points do not match exactly with the lognormal distribution. There is some distortion on the data point fit versus the lognormal distribution, but it was anticipated, as it illustrates system variation and characteristics. Another issue that affects the lognormal fit is that zero event days do occur in systems, yet the lognormal distribution cannot have zero values, so this censoring slightly affects the process of the fit. Zero values occur because system interruption data has discrete and continuous components, namely faults occur or they do not, yet SAIDI values are normalized and continuous in nature. Nevertheless, the

lognormal distribution is more adequate for this data than close competitors, Normal and Weibull distributions [53][53].

## 4.2 SAIDI/SAIFI

The next step after confirming the data's accuracy and distribution was to implement IEEE1366's processes and calculations. The first phase was to obtain reliability numbers for each year, which then would be used to find the $T_{MED}$ value for each year. With the MED thresholds, then the SAIDI and SAIFI results excluding the MEDs, could be calculated.

The implementation of the whole method was done in Matlab release 14. For sustained interruptions, only SAIDI and SAIFI were implemented for the reasons explained in section 3.4.2. No momentary indices were implemented since momentary data was not made available to the investigator. Load indices could not be implemented since interruption databases were based only on customer numbers. Additionally, any other sustained indices that were desired could be obtained in post processing, with little or no effort, which did not justify the implementation of them within the code. Following typical application, the only interruptions not included within the calculations were those of system operations outside the distribution system (transmission, plant, etc) and those of planned maintenance interruptions. All other faults were included in the calculations.

The interruption database did not include total feeder clients, only clients affected by a given interruption. Thus a client database for the feeders under study was created separately. The Matlab code was written in a way to be able to cross reference the total number of clients per feeder from the client database, while using the interrupted client data from the interruption database. In case there was feeder renaming, both feeder names were used, each with their client bases as specified at the time of interruption. If the program found that the affected clients were higher than the number of total clients in the feeder, the calculation was adjusted to reflect this. If more clients were interrupted, it would indicate that a normally open tie between feeders was closed at the time when the interruption occurred, thus the total client base at that time is different. Thus if this

happened, the client base was adjusted to the number provided by the interruption database. A sample code for the SAIDI calculation is provided below.

**Sample SAIDI Code:**

```
if reg_0(j,1)>=36526 && reg_0(j,1)<=36891
    if reg_0(j,12)>feederoutput(i+feederindex,4)
        clients=reg_0(j,12);
    else
        clients=feederoutput(i+feederindex,4);
    end
    if clients==0
        clients=99;
    end

    saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
    feederoutput(i+feederindex,10)=saididaily+feederoutput(i+feederindex,10); %SAIDI
    saididailynumonly=(reg_0(j,12)*reg_0(j,7));
    regdisoutput(i+feederindex,10)=saididailynumonly+regdisoutput(i+feederindex,10);

    if saididaily~=0
        MEDdatalog00(c,1)=log(saididaily);
        c=c+1;
    end
end
```

It is with this code that fault code extraction is also taking place. Each event is added to its respective SAIDI daily value and its fault code is stored to be used for weighting by the framework several phases later. SAIDI/SAIFI results are calculated on the feeder client base, area or region client base and system wide client base.

## *4.3 MED Threshold*

Major events can seriously affect reliability numbers, thus they are generally undesirable. Thus the need for a method to eliminate them is as important as the reliability numbers themselves. Using the method from IEEE1366 and detailed in 3.2.4, a script was added to the SAIDI/SAIFI code in Matlab.

The calculation of $T_{MED}$ for the 2.5Beta method requires data from five previous years to the period under study. The investigator had 7 years of accurate data, and one

previous year, were the data was not complete in many cases. This previous year was used to calculate the $T_{MED}$ for the first accurate database year. After that, all other periods kept accumulating years until the five requisite years were complete; this occurred on year 5. Major event thresholds ($T_{MED}$) were calculated using equation (3-41) included within the script. The use of the 2.5Beta methods in this phase was what required at a previous phase the verification of the skewness of the distribution, and confirmation of its log-normality.

The IEEE1366 does not specify at the level MEDs should be calculated, and is left open to the application. For this project, detailed feeder data was made available, so threshold values ($T_{MED}$) were calculated on a feeder basis, not on a system wide level. The $T_{MED}$ values were used to determine days exceeding normal operation bands — MEDs— for each feeder. The possibility of encountering more than 365 MEDs per year is acceptable and justified, since there could be a total number of $365 \times N_F$ MEDs –one per day for each feeder. Some sample code below for the calculation:

**Sample MED Code:**

```
meantempA=mean(MEDdatalogA);
devtempA=std(MEDdatalogA);
combotempA=meantempA+2.5*devtempA;
feederMEDtuki(i+feederindex,4)=exp(combotempA);
```

More detail on the process can be discovered by inspecting the full code in Appendix A.

## *4.4 Fault Code data*

Fault code data was extracted during the first phase of the SAIDI/SAIFI calculations. In the script, the fault code was extracted from the database when the event information was being read for calculation. It is stored in a separate file but maintaining its index association to the original event. This way any cross-reference required can be done, since the relationship to the feeder is not lost. A partial code sample is as follows:

**Sample Fault Collector Code:**

```
if reg_0(j,13)==51
    if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
        feederFAULT98(i+feederindex,8)=feederFAULT98(i+feederindex,8)+1;
    elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525
        feederFAULT99(i+feederindex,8)=feederFAULT99(i+feederindex,8)+1;
    elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891
        feederFAULT00(i+feederindex,8)=feederFAULT00(i+feederindex,8)+1;
    elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256
        feederFAULT01(i+feederindex,8)=feederFAULT01(i+feederindex,8)+1;
    elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621
        feederFAULT02(i+feederindex,8)=feederFAULT02(i+feederindex,8)+1;
    elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986
        feederFAULT03(i+feederindex,8)=feederFAULT03(i+feederindex,8)+1;
    elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352
        feederFAULT04(i+feederindex,8)=feederFAULT04(i+feederindex,8)+1;
    end
        feederFAULT(i+feederindex,8)=feederFAULT(i+feederindex,8)+1;
end
```

## 4.5 *Weighting Process*

The weighting step was the final phase added to the Matlab code. The process was described in section 3.4.2 but no implementation details were given; these will be provided in this section. The critical part of the framework is the extraction of the fault codes during the calculation of initial SAIDI/SAIFI values. When the program is reading the interruption data, duration, sector/region, feeder, and fault code entry is extracted. These fault codes are stored in a separate matrix which is roughly the same dimension as the SAIDI output matrix. The index that ties each fault code to each event is not removed, thus daily SAIDI values can be tracked to the individual fault codes which contributed to its result. The weighting step is divided in three main phases in four different programs (RIPPER, WEIGHTER, STRIPPER scripts and ADJUSTER function). Initial steps of data collection and storage matrix creation are part of the SAIDI/SAIFI program, as is integral to the calculation of the SAIDI/SAIFI numbers. A second phase where weights are calculated and stored is contained within two separate scripts, one Matlab function and one script. The final process, the recalculation of SAIDI/SAIFI numbers with the

framework adjustment is implemented in a final, third, script combining the steps from phase 1 and the weighting function from phase 2.

### 4.5.1 Weighting Process Phase 1

After initial data scrubbing and repair steps have been completed, SAIDI/SAIFI calculations can take place. Simultaneously, the first phase of the weighting takes place. Matlab script RIPPER calculates SAIDI daily values using the interruption data, and while this takes place, the fault codes are extracted. Processes are made in order of feeder, then in order of event by date and time. It is typical to find many SAIDI zero days at the feeder level, but once regional numbers are calculated, zero SAIDI days are rare. Typically, at feeder level only one interruption is seen per day, if any. However, if more than one interruption occurs, SAIDI values are accumulated to create the daily value. Each of the fault codes is counted and stored. Several counters are running simultaneously at this time. It will count the amount of each fault code for each feeder, plus the amount per area/region if so defined, and a system wide value. This is done to have several levels of resolution. The investigator did detect that the best level for analysis is at area level since there are more data points than at feeder level, yet not much area uniqueness is lost as in the cases of system wide or regional levels. Fault codes were maintained in yearly periods for each level.

### 4.5.2 Weighting Process Phase 2

Phase two consists of two main codes, a Matlab script and a function. The first execution of phase 2 is the script. The script WEIGHTER will process the fault data and interruption data from RIPPER and produce weight factors to allow the framework to adjust SAIDI at a later stage. WEIGHTER has several modes of operation as proposed in section 3.4.2. The modes are the following: PAIR, MEAN and MAXIMUM. The three modes are designed to produce numbers based on a slightly different weighting method from one another. Three were designed to allow for the evaluation of the most adequate one for the framework. All three are based on a similar method, and as so, were implemented to verify which provided better performance for the application at hand. For

all modes, WEIGHTER creates the weight matrix which will be used by ADJUSTER, the adjusting factor function used in phase 3. The complete code can be found in Appendix A.

#### *4.5.2.1 Mode 1 PAIR*

This mode is based on the pairing of fault codes to produce weights. Fault codes are read from the files created by RIPPER. The weight matrix is an array of size $A \times B$, where $A$ is the number of fault codes used and $B$ is the number of areas/regions/systems under study. A vector of $A$ values is also created. This vector is the *comparison vector*. Means are calculated for each fault code across the areas under study. The mean occurrence of each code across the system is stored on the comparison vector. The comparison vector values are then ranked from lowest occurrence to highest occurrence. This provides a rank for each fault code according to the cross-system mean values. A vector is then created from the fault codes from each system under study. This vector, the *system vector*, is also ranked the same way as the comparison vector. This is where the pairing process is completed. The most occurring code is paired to the least occurring code, and vice versa. The following equation provides the new rank for the system vector based on the ranks on comparison vector,

$$R = N - S_R \tag{4-1}$$

Where R is the new rank, N is the total fault codes and $S_R$ is the original system rank. The ranks are now paired, but no changes have been made to the weight matrix. The weight matrix contains the fault code in percentage of the total fault codes per system. The new rank for each code, derived from the pairing process is also normalized against its total, thus making, in essence, a comparison of 1 to 1 – both totals are the same but distribution is different. The ranks of the system in the weight matrix are now compared to those from the normalized comparison vector. If the fault code has a higher rank in the comparison vector, the percentage in the matrix is increased by the difference

between the two. Conversely, if it is lower, it is decreased by the same amount. This will be the factors applied by ADJUSTER.

#### *4.5.2.2 Mode 2 MEAN*

This mode is completed in the same way as PAIR, but no pairing takes place. The percentile ranks in the weight matrix are compared to the percentile ranks based on the cross-system mean values. These are then the factors used by ADJUSTER.

#### *4.5.2.3 Mode 3 MAXIMUM*

This mode is completed in the same way as MEAN. The percentile ranks in the weight matrix are compared to the percentile ranks of the maximum cross-system values instead of mean values. These are then the factors used by ADJUSTER.

### 4.5.3 Weighting Process Phase 3

Phase 3 involves the use of the ADJUSTER function in conjunction with a simplified RIPPER script – STRIPPER. STRIPPER is a RIPPER script but with no data scrubbing or fault code extraction, and all SAIDI daily values are modified by ADJUSTER. This is a sample call for ADJUSTER:

**Sample ADJUSTER Code:**

```
saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),
   inputoption);
saididailynumonly=(reg_0(j,12)*reg_0(j,7));
saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutputMEDDED(i+feederindex,2),
   reg_0(j,13),inputoption);
```

The adjusted SAIDI daily value is processed the same way as was done before without the adjustment. The mid-process adjustment is transparent to the IEEE1366

method. Yet the results vary considerably, as will be described in Chapter 5. The entire process for the framework is presented in Figure 4.5.1.

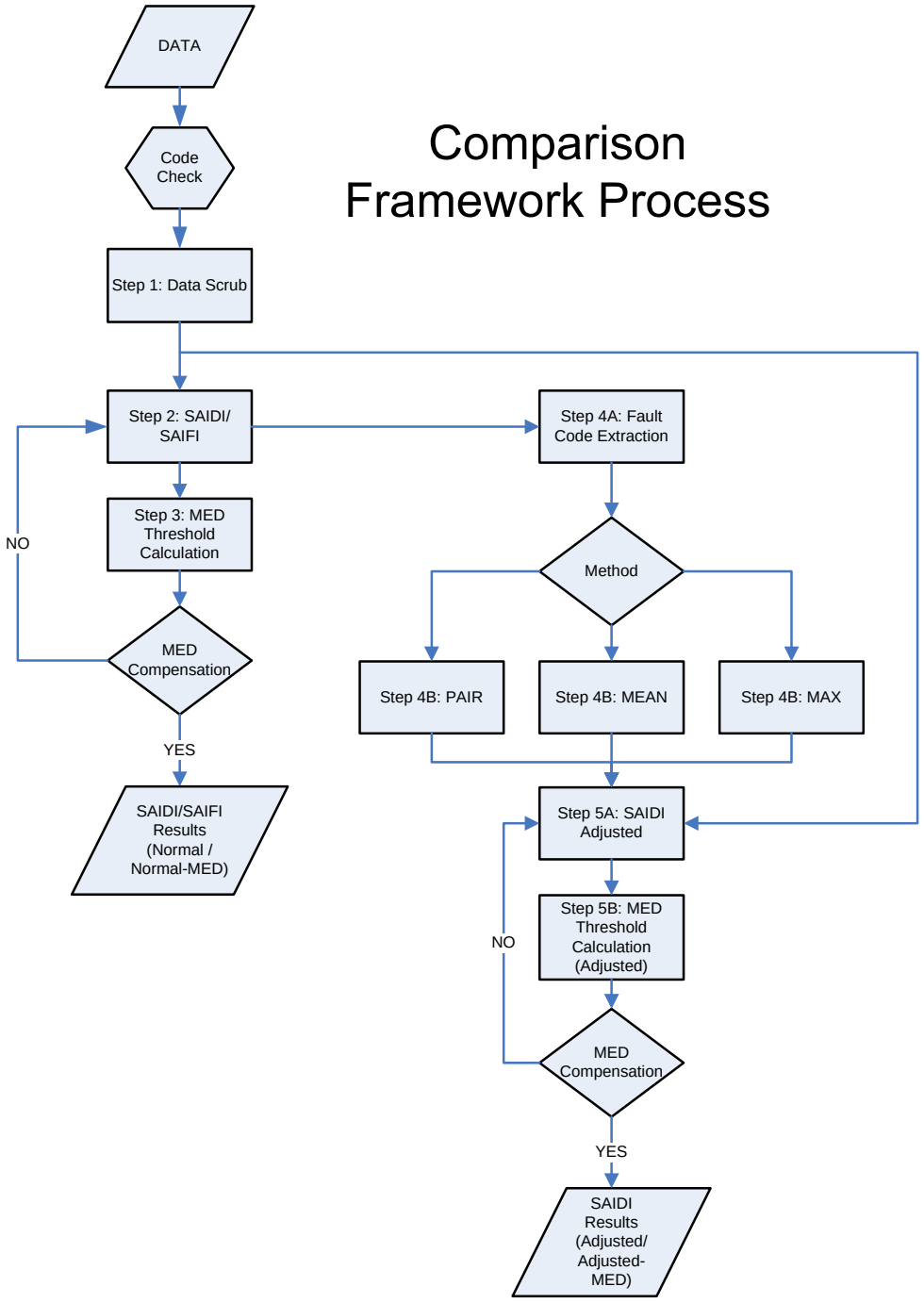



**Figure 4.5.1. Framework Process Flowchart**

## 4.6 Average Daily Normalized SAIDI

During the preliminary phases of the study, it was determined that a more efficient method for quantifying the effectiveness of the framework was needed. Using yearly SAIDI values for a study of considerable duration, such as this one, proved to be cumbersome at best, for the excessively large amount of indices being handled. In our case, this meant the comparison of 7 years worth of data for 25 areas. If using SAIDI alone, this required the direct comparison of 175 values — a more efficient approach was sought.

A composite index based on daily values for the entire duration of the study was determined to be the best option. The index developed for use was an a*rea average daily* SAIDI result *normalized* against the area client base. This value, the *Area Normalized Daily SAIDI* (referred to as *ANDS* from here on for simplicity) was used throughout the project to compare results. The value would, in other words, provide the average sustained interruption duration value for a client, on a daily basis. If this value was added for the total amount of days in the study period, it should equal the *mean daily* SAIDI value for the area or region. ANDS is obtained by calculating the mean (geometric mean in this case, due to the lognormality of the data) of the whole set (the entire study period) of SAIDI daily values for the area client base, while dividing it by the total client base. This is very efficiently calculated as an additive series in the Matlab code.

$$ANDS = mean\left(\frac{\sum Customer \quad Interruption \quad duration}{(Total \quad number \quad customers)\times(Entire \quad Period \quad Duration)}\right) \tag{4-2}$$

$$ANDS = mean\left(\frac{\sum r_i N_i}{N_T \times D}\right) \tag{4-3}$$

Where, $D$ is the duration of the period in coherent units with $r_i$. The relationship with SAIDI can be seen in Eq. (4-4).

$$ANDS = mean\left(\frac{SAIDI}{D}\right) \tag{4.4}$$

Note, that even though these would be the equations for calculating ANDS, it is also obtained by averaging (by calculating the geometric mean) all the SAIDI daily values during the entire period. This is one of the reasons for its efficiency – it uses part of the SAIDI calculation to derive its result. The use of this index instead of yearly SAIDI provided a faster and more direct way for the comparison of area performance. Since there is one constant value of ANDS for an area throughout the entire period, much less results need to be analyzed. In our case, a reduction of 175 indices to 25 was achieved. This accelerated the analysis considerably. It also gave a better idea of the daily impact of SAIDI and it was more directly comparable to the values used for the statistical fit of step 1. Additionally, it was simple to implement the math for this procedure in the Matlab code, since it used numbers which were required by the program to complete the SAIDI/SAIFI calculation process.

Since the seven years of SAIDI data were compressed into one index result for each area, yearly variation data is lost. This was considered during the development of the index, and a solution was found. Along with calculating the mean value for the ANDS, its variance is also calculated for the period. Thus the ANDS index provides the mean and apparent variation of the mean for each area. Thus, if variance data is also considered, a total set of 175 mean values and 25 variance results (one for each area) is reduced to 25 ANDS results and 25 ANDS variance values. During the study it was determined that the use of ANDS was far superior to yearly SAIDI values for comparing area data, since two areas (with and without framework adjustment) could be compared with 4 values (1 ANDS and 1ANDS variance per area) instead of 16 values (7 SAIDI and 1 SAIDI variance per area).

# CHAPTER 5 DISCUSSION OF RESULTS

## 5.1 *Reliability Index Calculation*

As described in the previous chapter, the framework process is composed of 5 steps. Step 1 consists on data cleaning and verification. In step 2, reliability indices are calculated; these results are the basis for subsequent steps. This step was completed by using real utility data made available to the author. The data included interruption and fault code information; momentary interruption data was not provided.

Step 2 required the use of the script RIPPER to produce SAIDI/SAIFI values and their mayor event day (MED) thresholds. SAIDI/SAIFI values with MED compensation are also computed in this process, along with fault code extraction. In average, anywhere from 311 to 2023 total MEDs per year (on a per-feeder basis-see section 4.3) were extracted from the daily interruption data. Extraction of MEDs changes SAIDI and SAIFI values up to 82% from non-MED compensated results in this study. In step 5, the weight adjustments were completed with and without MED compensation and the results are presented in a subsequent section. For the entire dataset, the skewness of the lognormal distribution for the different areas was in the range of 21.35 to 166.52.

## 5.2 *Fault code use and impact*

As detailed in chapter 4, fault code extraction was critical for the implementation of the framework; this was completed in Steps 2 and 3. As part of the project, fault type, cause and occurrence statistics were recorded and analyzed. Seven years of data were made available to the investigator. Fault causes were studied and consolidated in common categories, using data from references [23],[49] and [52], and system data made available to the author. Average percentages for fault causes are provided in Figure 5.2.1.

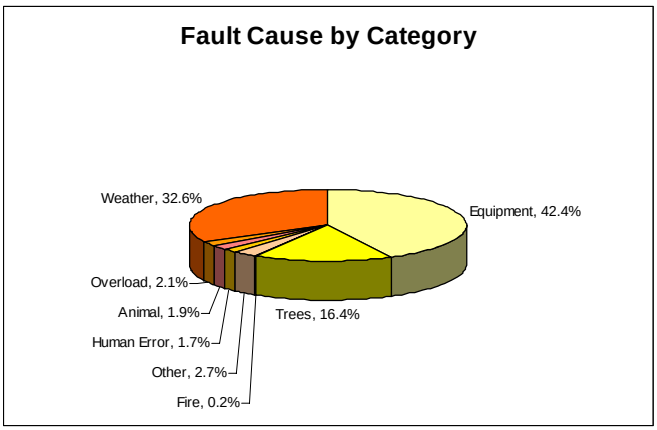



**Figure 5.2.1. Fault Causes by Percentage**

The problem that arose in section 3.3.2 reappeared when comparing results from the references: diverging categories and definitions. Nevertheless, it can be observed that this data is close to some of the categories in the references. Tree contact percentage observed in the three referenced studies is very close and within a range of 10-19%. Animal contact is considerably less, by a factor of ten from those of the references. Weather issues are higher than those seen in the referenced studies. Nevertheless, the main difference is the large contribution from equipment failures. This is in serious contrast with the references. It must be mentioned that equipment failures may be numerically disadvantaged against other type of faults and might no be as high at it appears upon preliminary inspection –this is explained next.

Many temporary interruptions, which should have been attributed to other main categories, are dropped into "catch-all" categories -such as *unknown*- due to bad data entry. This decreases the number of temporary interruptions by moving faults to categories that attract little to no attention. Permanent faults (i.e. equipment failure), unlike temporary faults, rarely go unrecorded or misreported. Temporary faults are, in general, not as severe as permanent faults, thus less attention is paid to them and their data recording. Thus, more entry error (and less fault data entry) is expected on temporary fault types than on permanent types. As with the other tables indicating fault causes, this one must also be approached with caution. Fault codes can have data entry

and definition ambiguity errors, *unknown/other* cause categories can contain faults from a number of categories -yet are not detected as such- and not recorded [55].

Permanent faults are generally well documented, and equipment failure faults are no exception to this rule. Equipment failures according to the studied data are divided as follows (See Table 5.2.1):

**Table 5.2.1. Equipment failure percentages**

| | |
|---|---|
| *Hazardous material/equipment removal* | *>1%* |
| *Transformer failure* | *2%* |
| *Pole/Structure failure* | *7%* |
| *Broken Overhead Conductor* | *49%* |
| *Arrester Failure* | *5%* |
| *Insulator Failure/Breakage* | *5%* |
| *UND Cable accessory failure* | *2%* |
| *UND Cable failure* | *12%* |
| *Failed switch* | *2%* |
| *Broken supports* | *1%* |

Faults categorized by the client base affected were also determined. As expected, partial feeder faults are the ones with most occurrences. Exactly half of all faults are of partial character and affect only a portion of the feeder client base. Total feeder and substation faults are evenly divided, with substation faults slightly inching total feeder faults in the percentages. Total feeder faults affect all clients in the feeder circuit, while substation faults affect all clients on all the circuits connected to the substation bus. Fault type percentages by customer base affected is presented in Figure 5.2.2:

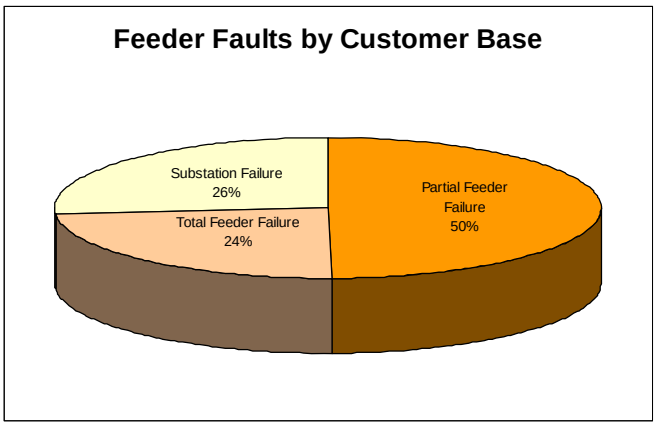



**Figure 5.2.2. Faults by Feeder Client Base Affected**

## 5.3 *Framework*

The comparative framework begins adjustments with the script STRIPPER. This takes place after SAIDI/SAIFI values have been calculated and compensated for MEDs in steps 2 (SAIDI/SAIFI) and 3 (MED Threshold), and fault codes extracted in step 4 (Fault Code Data). The objective of the framework was to design a method that could strip the interruption data of some of its uniqueness to provide a leveled comparison field for the evaluation of SAIDI/SAIFI data from distinct systems or system areas.

In order to verify that the process worked as expected (described in section 4.5), real interruption data from utilities was solicited. Data from very different service areas was obtained, including:

- Urban
  - Commercial/Downtown underground
  - Residential underground
  - Residential overhead
  - Industrial overhead
- Rural
  - Residential overhead

It must be mentioned, that even though certain areas are considered mainly one type of configuration, it may have any other type of load or configuration. When an area configuration is mentioned, it must be kept in mind it is *mostly* that kind of load/topology setup.

### 5.3.1 Framework RIPPER results

To qualify and quantity the success of the framework, the process was implemented in Matlab Release 14 (V7.0). For comparison purposes, yearly SAIDI/SAIFI and a composite daily SAIDI/SAIFI value was developed for this project. The composite value is obtained by calculating SAIDI values for an area in a daily process. It gives the *area average daily* SAIDI result *normalized* against the area client base. This value, the *Area Normalized Daily SAIDI* (referred to as *ANDS* from here on for

simplicity) was used throughout the project to compare results. This index is not part of the IEEE1366 standard, but developed in this project and recommended for similar studies. The details of this index were presented in section 4.6.

The data provided to the author was for 25 distinct areas within a large distribution system. These areas were assigned names Alpha through Tango, from the military phonetic alphabet. All analysis was completed using these artificial name tags, to avoid any unintentional biasing of the data.

It must be mentioned that the data acquired by the author was for one major system. The areas in the study are parts of this greater system, nevertheless the framework is expected to work adequately with comparison among entire systems, as long as sample spaces are adequate. The larger the system under consideration, the larger the sample space required to provide a good idea of system performance. This is required due to the fact that consolidating large amounts of data can lead to a homogenization of sample spaces. For the framework it is recommended to use data divided by regions within these larger systems to aid the analysis. This should also provide better resolution for the analysis. The amount of regions within the systems under study will not affect the results if fault codes are coherent between systems, as specified in chapter 4.

Plots in this chapter follow a consistent labeling. Legend entries in the plots indicate the following:

- ***Normal:*** SAIDI/SAIFI/ANDS values with no framework adjustment. MED extraction may or may not have taken place. This is indicated in the title and suffix.
- ***Adjusted:*** SAIDI/SAIFI/ANDS values with framework adjustment. MED extraction may or may not have taken place. This is indicated in the title and suffix.
- ***MED:*** Suffix appended to legend entries which indicated MED compensated values. Both *Normal* and *Adjusted* values can be accompanied by this suffix.

Two main kinds of plots are presented in this chapter. One is yearly variation, the other is area variation:

- ***Yearly Variation***: These plots present SAIDI, SAIFI or MED values with independent axis as yearly timeline. The dependent axis is SAIDI or SAIFI and is provided in percentile against the set maximum, unless indicated.
- ***Area Variation***: These plots present ANDS values for the different areas with independent axis as area. The dependent variable is ANDS in percentile against set maximum, unless indicated.

The ANDS results for steps 2 and 3 are presented in Table 5.3.1 and Figure 5.3.1 when no MED compensation takes place.

**Table 5.3.1. ANDS results: All Areas without adjustment or compensation**

| *Alpha* | *Bravo* | *Charlie* | *Delta* | *Echo* | *Foxtrot* | *Golf* | *Hotel* | *India* |
|---|---|---|---|---|---|---|---|---|
| *0.62403* | *0.71148* | *3.6698* | *0.91434* | *2.562* | *1.7471* | *0.71322* | *0.6408* | *1.1209* |
| *Juliet* | *Kilo* | *Lima* | *Mike* | *November* | *Oscar* | *Papa* | *Quebec* | *Romeo* |
| *0.93524* | *0.98406* | *0.42792* | *0.97727* | *0.8833* | *1.177* | *0.64543* | *1.5092* | *1.9951* |
| *Sierra* | *Tango* | | | | | | | |
| *2.4514* | *0.80397* | | | | | | | |

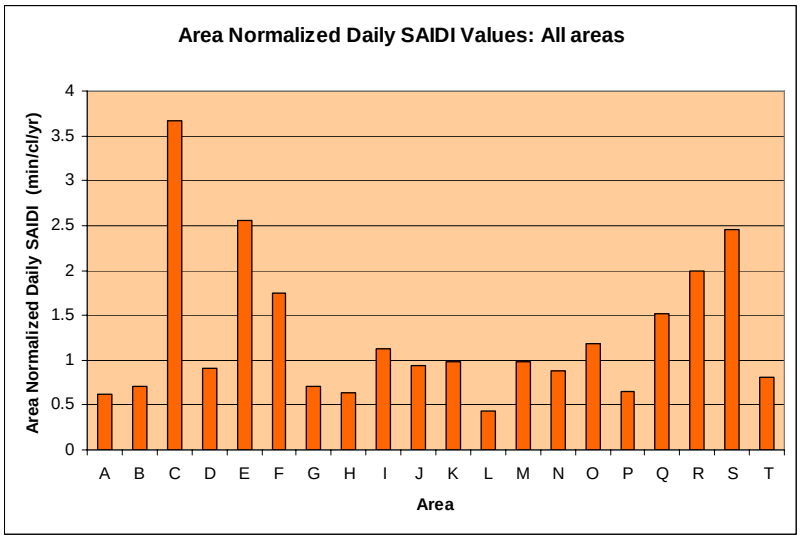



**Figure 5.3.1. ANDS values: No Adjustment or Compensation**

Observing Figure 5.3.1, it can be concluded that ANDS values are all over the map. It can also be expected for SAIDI values to behave similarly. SAIDI and ANDS are

tightly related. ANDS is a normalized version of the standard SAIDI index. Thus, even though final numerical values are different, and SAIDI values are more than those of ANDS (ANDS has one value per area, SAIDI has 7 per area in this study), the trends will be same when averages are calculated. SAIDI will have one value per year, while ANDS will have one value per study period. The average of SAIDI for the entire study period will follow the exact trend of ANDS, even though numerical results are different. In percentile, the results are equal. The percentile difference from the lowest ANDS value, Lima, to the highest, Charlie, is 856%; the variance is considerable. Not only are values quite different from each other, but comparing SAIDI for consecutive years for each area, presents similar issues. Figure 5.3.2 presents SAIDI variation for Lima area in a yearly base.

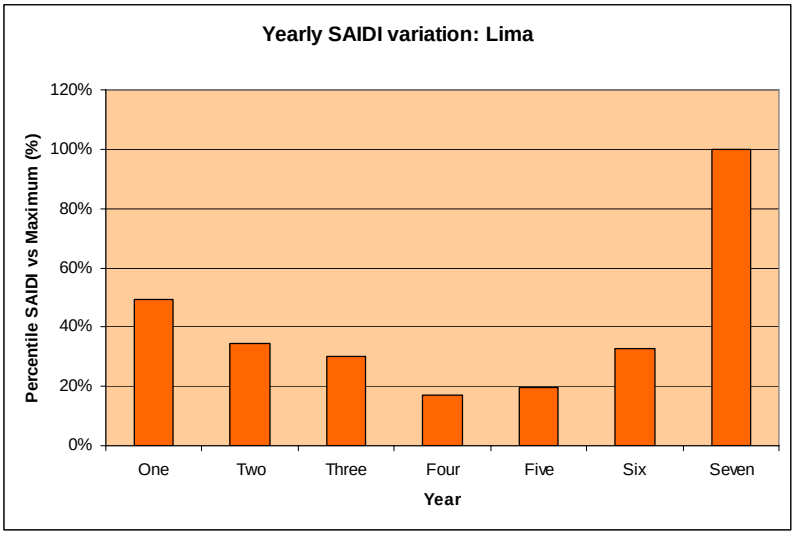



**Figure 5.3.2. Yearly SAIDI Variation: Lima Area**

It is clear that SAIDI values are not precise numbers. They vary from season to season and year to year, as expected and discussed in section 3.3. Nevertheless, trends are present in the results. Yearly comparative performance for Charlie and Lima areas are presented in Figure 5.3.3.For example, Charlie area was -on average- the worst performing area with respect to SAIDI numbers, while Lima was the best. This trend is apparent comparing both areas in yearly plots. It can be seen that year 7 was very bad for performance.

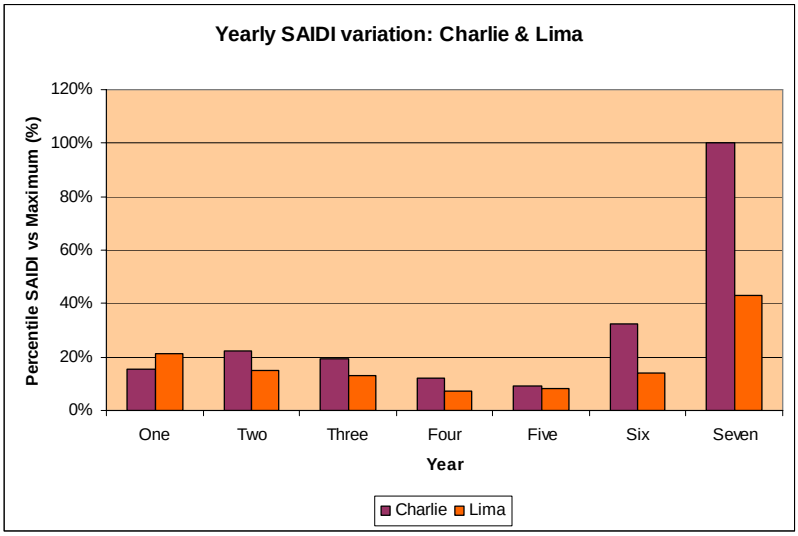



**Figure 5.3.3. Yearly SAIDI Variation: Charlie & Lima Areas**

In all but one year, Charlie area has higher numbers for SAIDI than Lima area. Thus, even though numbers vary from year to year, *relative* performance can be predicted. Another trend is apparent: yearly variation is *similar* for both areas. This may or may not occur within areas and can be tied to the peacefulness of the reporting period. A peaceful year is one where little or no major events occur -no storms, hurricanes or the like. By contrast, a restless year or period is one with many major events. This can be confirmed by evaluating the *variance* for each of the reporting periods. A high SAIDI variance is a telltale sign that major events could have happened. It might also mean that operational differences have been made, but all things remaining equal, high variance would most likely be due to uncontrollable events, i.e. major events. Figure 5.3.4 presents total major events extracted on a yearly basis for all areas.

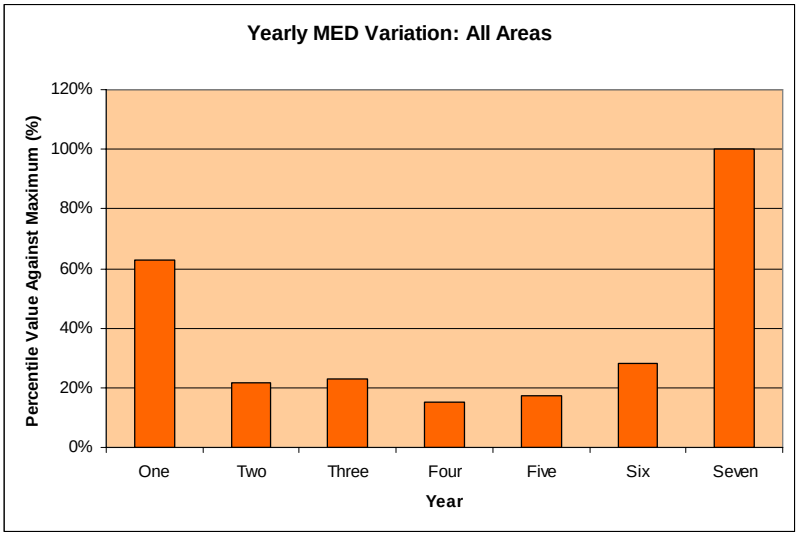



**Figure 5.3.4. Yearly MED Variation: All Areas**

Even though there is no direct shadowing of figure 5.3.4 to 5.3.3, a general decreasing tendency from year 2 to year 4 is also present in Figure 5.3.4, while an increasing one is seen from year four onward. Figure 5.3.5 presents the yearly systemwide average of SAIDI variation. It can be seen that it closely follows the trends of Figure 5.3.4.

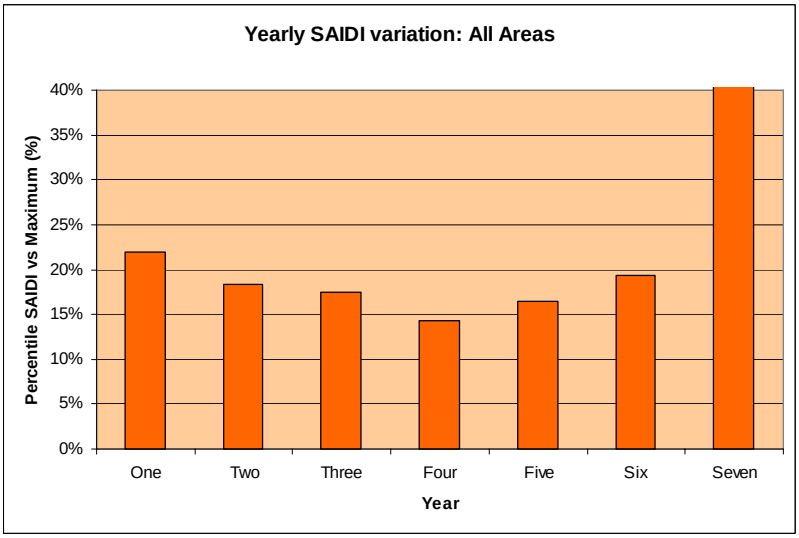



**Figure 5.3.5. Yearly SAIDI variation: All Areas**

The percentile axis was adjusted to allow for better resolution, year 7 is 100%. Year 7 experienced severe storms, which explains the spike in SAIDI numbers. This figure presents very convincing evidence that yearly area variation follow a certain trend. Nevertheless, there is serious variation between areas. This effect is somewhat softened when systemwide averages are computed, yet trends are still visible. The effect of the extraction of MEDs is presented next.

### 5.3.2 Major Events

Major events reflect as outliers in system performance plots and result in reliability numbers that are very far from the actual average. SAIDI is highly affected by MEDs, even though they affect SAIFI in a limited extent [15][56]. The extraction of MEDs allows the mean value of SAIDI for a given area to decrease since outliers are taken out of the calculation. The effect of this reduction is presented in Table 5.3.2 and Figure 5.3.6.

**Table 5.3.2. ANDS results for All Areas; MED Compensated**

| *Alpha* | *Bravo* | *Charlie* | *Delta* | *Echo* | *Foxtrot* | *Golf* | *Hotel* | *India* |
|---|---|---|---|---|---|---|---|---|
| *0.45263* | *0.49557* | *2.8374* | *0.54238* | *1.4334* | *1.2242* | *0.37882* | *0.48577* | *0.78484* |
| *Juliet* | *Kilo* | *Lima* | *Mike* | *November* | *Oscar* | *Papa* | *Quebec* | *Romeo* |
| *0.66049* | *0.51716* | *0.3511* | *0.75835* | *0.68527* | *0.94881* | *0.47924* | *1.2873* | *1.5993* |
| *Sierra* | *Tango* | | | | | | | |
| *2.2187* | *0.66268* | | | | | | | |

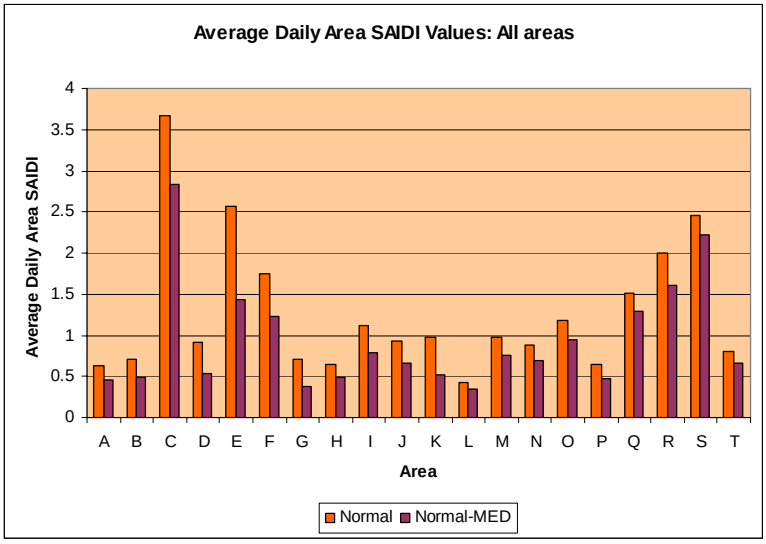



**Figure 5.3.6. ANDS: All Areas Normal/Compensated Comparison**

Comparing Tables 5.3.1 and 5.3.2, it is clear that MED extraction lowers SAIDI numbers in all areas; this is also visible in Figure 5.3.6. The extraction of MEDs reduces the variance of results by eliminating outliers. This shifts means toward zero, or more precisely, toward the true system performance mean. In a SAIDI yearly variation plot, the effect of the MED extraction on outliers is clear. This is presented in the Figure 5.3.7.

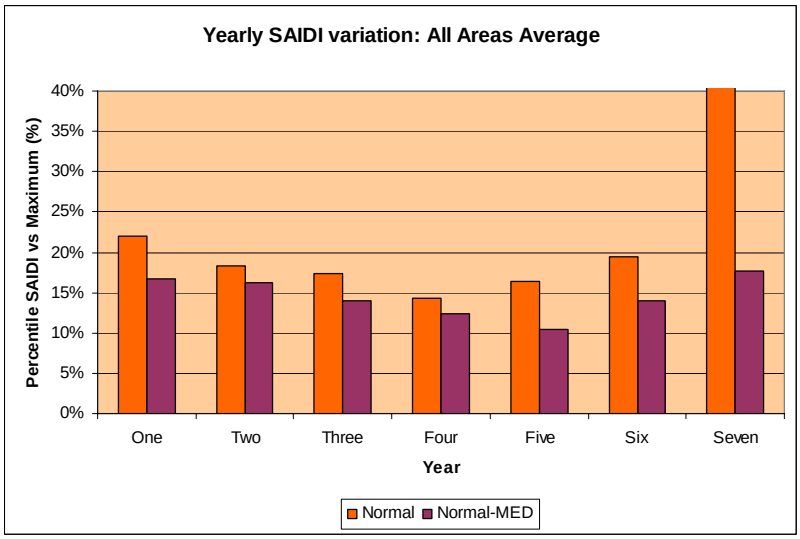



**Figure 5.3.7. Yearly SAIDI: All Areas Normal/Compensated Comparison**

From Figure 5.3.7, it is clear that MED extraction does influence results greatly. Comparing normal and compensated values in the figure, the reduction of SAIDI for year 7 is visible (percentile axis was adjusted for resolution; year 7 normal value is 100%). SAIDI is reduced on all years with compensation, especially in year 7; this is due to the great number of outliers (extracted outliers: 2023) on year 7. The removal of outliers provide better resolution for the study as can be seen when comparing normal and compensated results.

SAIFI on the other hand is only moderately affected as described in reference [3]. This is due to the fact that SAIDI is greatly increased in major event situations due to severely increased outage durations [47][48]. Since SAIFI does not measure time of outage, only incidence, a short duration is the same as a long one. If the major event does not cause a large number of events, SAIFI will not be severely affected; it is natural that MED extraction will not impact SAIFI as it does SAIDI. Figure 5.3.8 presents this fact.

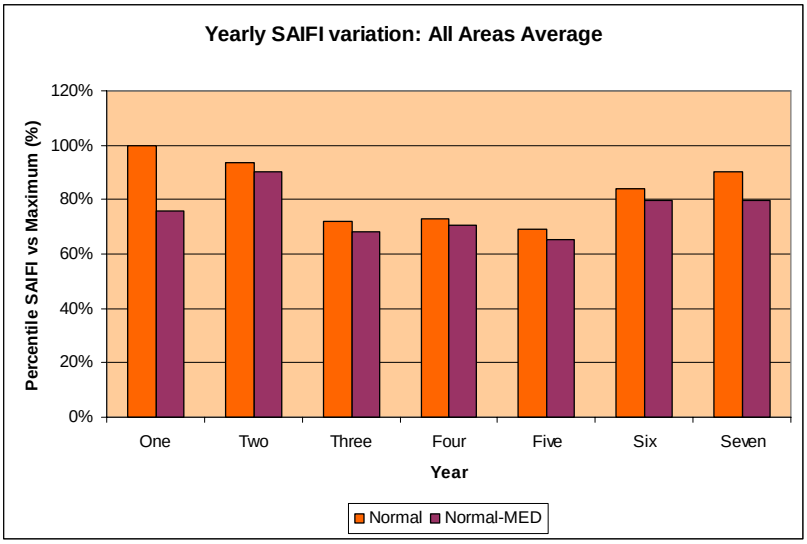



**Figure 5.3.8. Yearly SAIFI: All Areas Normal/Compensated Comparison**

The comparison framework was implemented for both non-compensated and MED compensated SAIDI/SAIFI processes. The framework adjustment was based on the process presented in Chapter 4. SAIFI values were not adjusted as specified in the previous chapter.

## 5.3.3 Framework WEIGHTER results

This section will describe and discuss the results produced by WEIGHTER script. In the final incarnation of WEIGHTER, three methods created the weights for the process. All three were based on the same underlying principle: the use of fault codes to effectively equalize data. As presented in Chapter 4, the three methods were developed to produce weights for the framework: PAIR, MEAN and MAX. A brief explanation will be repeated here for convenience, but detailed description of the methods was presented in chapter 4. The methods and their description are as follows:

- ***PAIR:*** weighting is completed by pairing most to least occurring fault codes for each system.
- ***MEAN:*** weighting is completed by comparing each fault code to the cross-system mean value for that fault code.

- ***MAX:*** weighting is completed by comparing each fault code to the cross-system max value for that fault code.

Using the three methods, the script produces a *comparison vector* to be applied to each area as described in chapter 4. The effect on the area would depend on how different the area fault profile is to the comparison profile. This is the base for the equalization. It varies for each area and for each interruption. The comparison profiles produced by each method are tabulated in Table 5.3.3:

**Table 5.3.3. Fault Code Weights**

| ***Fault Code and Assigned Weights*** | | | |
|---|---|---|---|
| ***Code*** | ***Pair*** | ***Max*** | ***Mean*** |
| *1* | *0.18981* | *0.000465* | *6.73E-05* |
| *2* | *0.004932* | *0.10409* | *0.096683* |
| *3* | *0.002462* | *0.016961* | *0.009503* |
| *4* | *0.009503* | *0.004795* | *0.005343* |
| *5* | *0.011145* | *0.011164* | *0.016908* |
| *6* | *0.001421* | *0.0925* | *0.11984* |
| *7* | *0.012444* | *0.093323* | *0.095432* |
| *8* | *0.000356* | *0.010306* | *0.011308* |
| *9* | *6.73E-05* | *0.011558* | *0.013433* |
| *10* | *0.013433* | *0.007801* | *0.005312* |
| *11* | *0.016908* | *0.050991* | *0.030171* |
| *12* | *0.096683* | *0.003793* | *0.004932* |
| *13* | *0.0019* | *0.00229* | *0.002462* |
| *14* | *0.000511* | *0.016639* | *0.023704* |
| *15* | *0.11984* | *0.003077* | *0.0019* |
| *16* | *0.008181* | *0.011701* | *0.015752* |
| *17* | *0.006772* | *0.001575* | *0.001421* |
| *18* | *0.030171* | *0.007049* | *0.00961* |
| *19* | *0.005343* | *0.017963* | *0.011145* |
| *20* | *0.023704* | *0.016138* | *0.012444* |
| *21* | *0.00961* | *0.14471* | *0.18981* |
| *22* | *0.095432* | *0.000394* | *0.000356* |
| *23* | *0.011308* | *0.041723* | *0.051571* |
| *24* | *0.015752* | *0.16807* | *0.076181* |
| *25* | *0.003903* | *0.001252* | *0.000511* |
| *26* | *0.002889* | *0.13286* | *0.17026* |
| *27* | *0.051571* | *0.006835* | *0.006772* |
| *28* | *0.17026* | *0.004616* | *0.002205* |
| *29* | *0.076181* | *0.009375* | *0.008181* |
| *30* | *0.002205* | *0.002398* | *0.002889* |
| *31* | *0.005312* | *0.003578* | *0.003903* |

It can be appreciated from Table 5.3.3 that MAX and MEAN methods produce very similar numbers. This outcome was anticipated since both are, in essence, the same, the difference between the two is the value used for normalization – the dataset mean or maximum. The pairing method produces considerably different results from the other two methods. This will be apparent when framework adjustment results are presented.

The comparison weights are adjustable and variable for the study dataset, so these values are unique to these areas and time period. Any change in the fault codes, period length or areas will cause regeneration of these numbers through the execution of the program. Additionally, the amount of weights is directly proportional to the amount of fault codes under consideration. It could be increased or decreased from this number; this would need to be evaluated for each study. Figure 5.3.9 presents final weights for areas Charlie and Lima. These are obtained by comparing the comparison profile to actual fault codes from the area. Note that they vary considerably for several codes and are unique to the area.

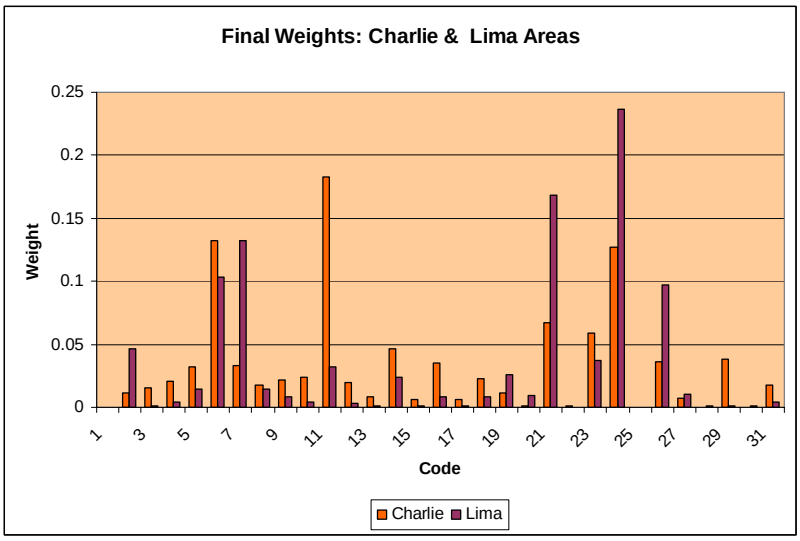



**Figure 5.3.9. Final ADJUSTER Weights: Charlie & Lima Areas**

### 5.3.4 Framework STRIPPER Results

STRIPPER is the final script for the framework and incorporates functions from RIPPER and the ADJUSTER function. ADJUSTER applies the weights determined by the previous steps to the interruption data, while the script is calculating system SAIDI. The adjusted SAIDI values for all areas (no MED compensation) along with normal numbers (also without MED compensation) are presented in Figure 5.3.10.

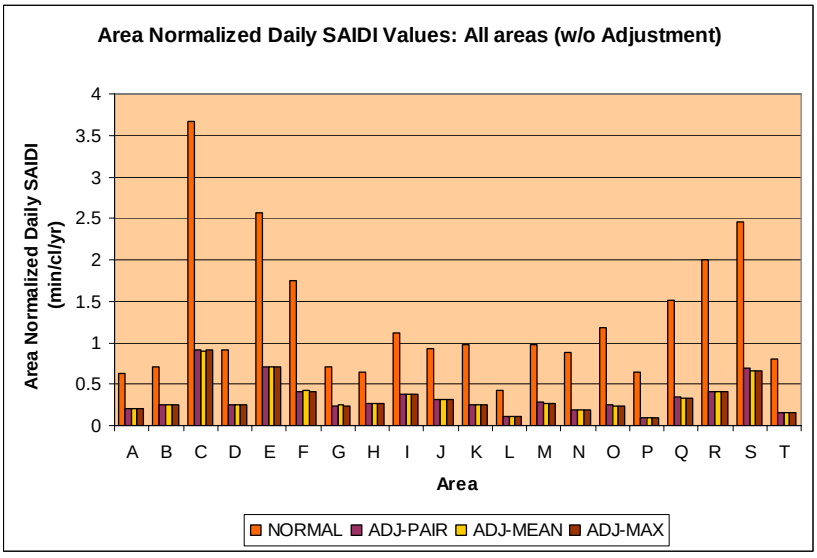



**Figure 5.3.10. ANDS: All Areas Adjustment/No Compensation Comparison**

Three things can be observed from Figure 5.3.10, namely:

- Adjusted ANDS values are consistently lower than non-adjusted values
- The results from the three framework methods is very similar
- Variation *among* areas is less with adjusted values

These results can also be observed if analyzing yearly SAIDI values instead of ANDS; Hotel area results (percentile axis adjusted, year 7 is 100%) are presented in Figure 5.3.11.

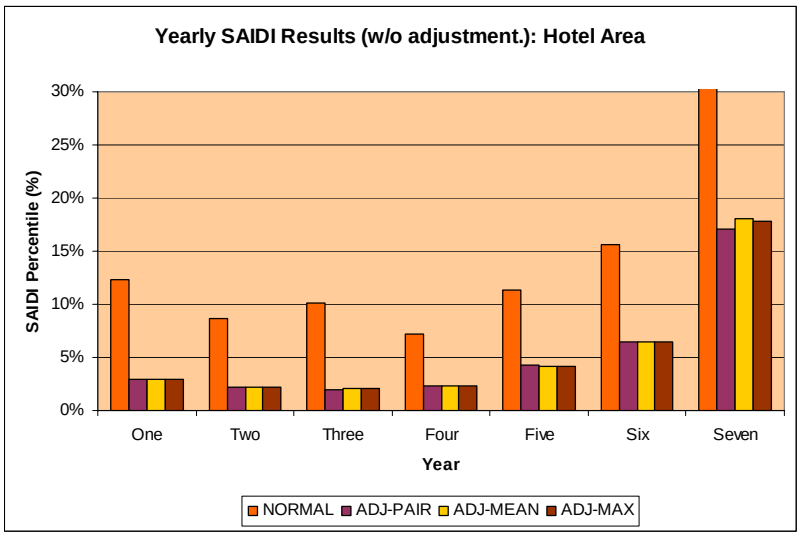



**Figure 5.3.11. Yearly SAIDI: Hotel Area Normal/Adjusted Comparison**

The three framework methods' results are very similar, amounting to no more than a normalized standard deviation of 0.022 between methods, with the biggest difference resulting from Sierra area. The similarity between the methods is also evidenced in Figure 5.3.12, a stack diagram of ANDS results. It can be seen that each of the methods produces about a third of the total, thus indicating that they all have similar results.

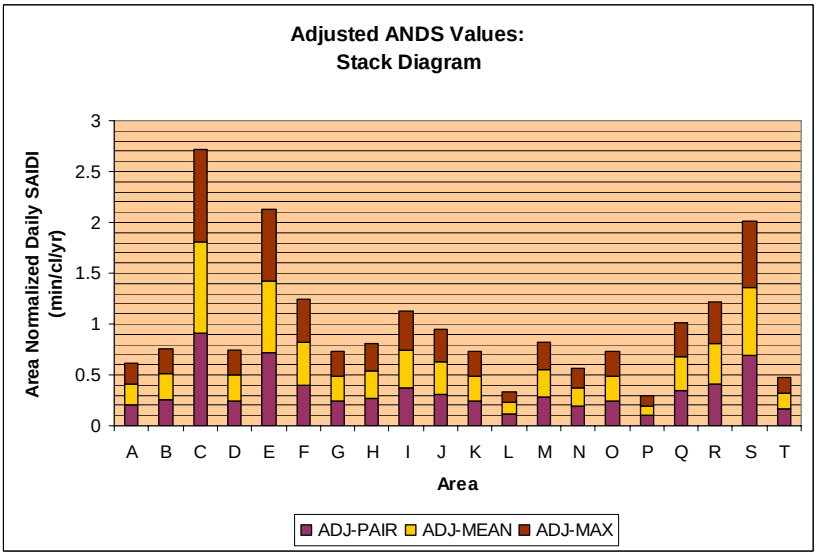



**Figure 5.3.12. Stack Diagram ANDS**

Yearly systemwide SAIDI values are presented in Figure 5.3.13. Normal (raw) SAIDI and their MED compensated counterpart values are included. Additionally, normal and MED compensated values produced through the framework process are also presented. These are adjusted-normal and adjusted-MED compensated values. Again, percentile axis was adjusted for resolution, with year 7 being 100%.

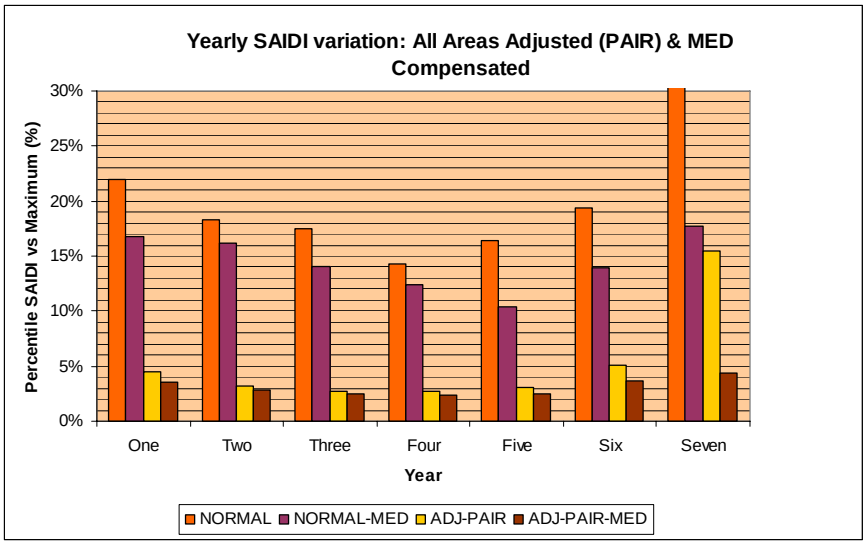



**Figure 5.3.13. Yearly SAIDI Variation: All Areas Adjusted/Compensated Comparison**

Notice that MED compensation for both sets (normal and adjusted) decreases final SAIDI, but in different amounts. With the normal dataset, MED compensation ranges from 1.9% to 82.3%, with a mean change of 5.2%. The adjusted dataset has a range of 0.3% to 11.1% with a mean change of 0.5% -a factor of ten. The framework causes this by decreasing the variance of the data. With less variation within the areas, less variation results in the systemwide numbers. When compensation takes place, less effective change in the final average value is obtained. As before, the framework produces lower numbers throughout, as seen with the ANDS. Yearly variation is also less than the normal dataset. For the normal dataset, normalized standard deviation between years, is 170% without MED compensation and 18% otherwise; 145% versus 27% in the adjusted dataset.

An interesting point is that the deviation for the adjusted values with compensation is higher than its analog in the normal set. This result was initially unexpected but later determined to be due to the framework itself. Since interruptions are

being weighted and variance is decreased by the framework process, some events which might have exceeded the $T_{MED}$ value without the weighting now do not exceed it. Thus less days are being extracted as MEDs, thus slightly higher standard deviation in the adjusted/compensated dataset is justified. MEDs removed after adjustment, range from 156 to 1464; an average reduction of 38.8% from those obtained with the normal set.

Variance in general, decreases with the application of the framework. This can be seen from Figure 5.3.14, which presents normal and adjusted results.

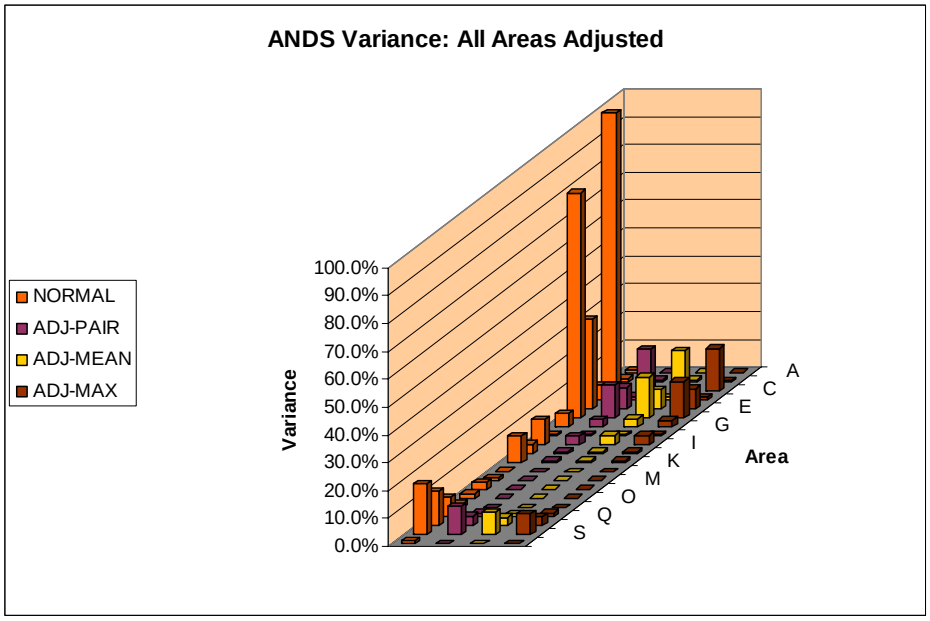



**Figure 5.3.14. ANDS Variance: All Areas Adjusted/No Compensation Comparison**

Careful inspection of this figure will provide some interesting information: the framework methods provide variance values that are not so similar between themselves unlike the case with ANDS. Additionally, not all variance is decreased from the value of the original dataset. Table 5.3.4 was prepared to illustrate this. Only four areas were included due to space constraints, but these are representative of the analysis. The four areas inspected more carefully were: Charlie, Golf, Hotel and Lima.

**Table 5.3.4. ANDS Variance: Adjusted/No Compensation**

| | *Charlie* | | *Golf* | | *Hotel* | | *Lima* | |
|---|---|---|---|---|---|---|---|---|
| | ***Var.*** | ***% Δ*** | ***Var.*** | ***% Δ*** | ***Var.*** | ***% Δ*** | ***Var.*** | ***% Δ*** |
| ***Normal*** | *1022.6* | | *50.946* | | *6.7306* | | *2.2258* | |
| ***ADJ-PAIR*** | *153.94* | *85%* | *28.337* | *56%* | *6.886* | *-2%* | *0.69097* | *69%* |
| ***ADJ-MEAN*** | *148.62* | *85%* | *30.264* | *59%* | *7.3304* | *-9%* | *0.70006* | *69%* |
| ***ADJ-MAX*** | *156.42* | *85%* | *25.913* | *51%* | *7.3382* | *-9%* | *0.54947* | *75%* |

Charlie, Hotel and Lima were selected because they are the areas with highest initial variance, increased final variance and lowest initial variance, respectively. Golf was added for comparison purposes. Charlie and Lima areas had similar variance reductions, even though they had very different initial levels. This change is expected, since it is not the original variance value which determines how much change will result, but the fault profile of the area. Charlie and Lima had different fault profiles, but the weights had similar properties, especially with high ranking codes. Hotel is the most interesting of the three, instead of lowering the variance, its final value is *higher* than its initial result. This was not entirely unexpected, but the reason was analyzed and justified.

The increase in variance is a justified outcome of the framework. The framework seeks to equalize characteristics to create a more leveled evaluation field for reliability studies. Thus it is expected that very stable areas will experience an increase in variability and/or mean values as the weights re-compute indices in a more equal form. High variance areas will see reductions, while low variance areas will see the opposite. This is only a general rule, it is not the initial variance level which determines the final result; fault profiles do.

MED compensation also took place for the adjusted values, and its variance impact plot is presented in Figure 5.3.15.

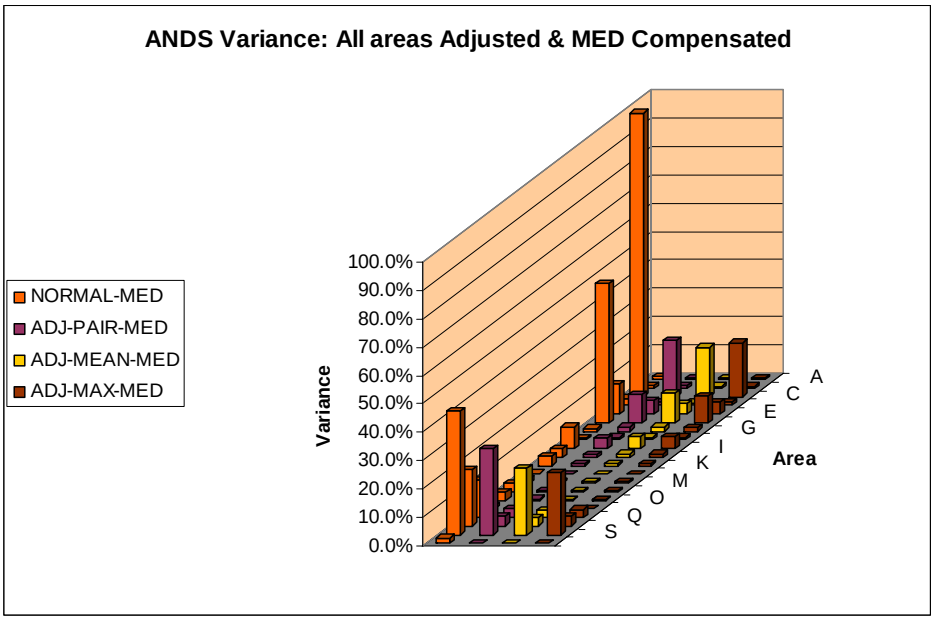



**Figure 5.3.15. ANDS Variance: All Areas Adjusted/Compensated Comparison**

It can be concluded that results are similar to the results without MED compensation. One important point that must be made here is that, not the same MED thresholds were used for normal and adjusted values, for obvious reasons. If normal major event thresholds were used throughout the study, no MEDs would have been extracted in the framework step, due to their high value. Thus, for adjusted values, equally adjusted thresholds were used. This would keep the same relative difference between a good and a severe value as in the normal dataset. Again, with the MED compensation, the areas were inspected for their variance results. As before, the representative areas were selected and tabulated in Table 5.3.5.

**Table 5.3.5. ANDS Variance: Adjusted/Compensated**

| | ***Charlie*** | | ***Golf*** | | ***Hotel*** | | ***Lima*** | |
|---|---|---|---|---|---|---|---|---|
| | ***Var.*** | ***% Δ*** | ***Var.*** | ***% Δ*** | ***Var.*** | ***% Δ*** | ***Var.*** | ***% Δ*** |
| ***Normal*** | *343.13* | | *2.3573* | | *3.39* | | *1.0887* | |
| ***ADJ-PAIR*** | *68.497* | *80%* | *2.8851* | *-22%* | *5.3363* | *-57%* | *0.39424* | *64%* |
| ***ADJ-MEAN*** | *59.73* | *83%* | *2.9341* | *-24%* | *5.4226* | *-60%* | *0.39077* | *64%* |
| ***ADJ-MAX*** | *65.049* | *81%* | *2.9651* | *-26%* | *4.3232* | *-28%* | *0.31121* | *71%* |

Instead of one area with increased variance, two were found, and these were integrated into the table along with areas Charlie and Lima, which, once again, were the

highest and lowest samples. From the table, we can observe that percentile change for Charlie and Lima are comparable to results from the non-compensated run. Nevertheless, Hotel has a much higher upward change, while Golf has half as much upward trend as Hotel – a total trend inversion from the previous run (51-59% change). To further justify the reason for the upward moving variances of areas Golf and Hotel, further analysis took place.

One of the reasons for the upward trend of areas Golf and Hotel is the weighting of the fault codes. If an area has a negative profile in comparison to the comparison profile, it will be greatly impacted by the process. This is due to the equalizing objective of the method. Since the area has such a different profile from the general population it will be *molded* to the general profile. How much it will be modified in accordance to the comparison (general) profile, will depend on two things: the client base of the area versus the total client base and the percentile difference between the areas fault codes and comparison fault codes. This can be observed in Figures 5.3.16 and 5.3.17, which present the comparison profile versus the fault code profiles for Golf and Hotel.

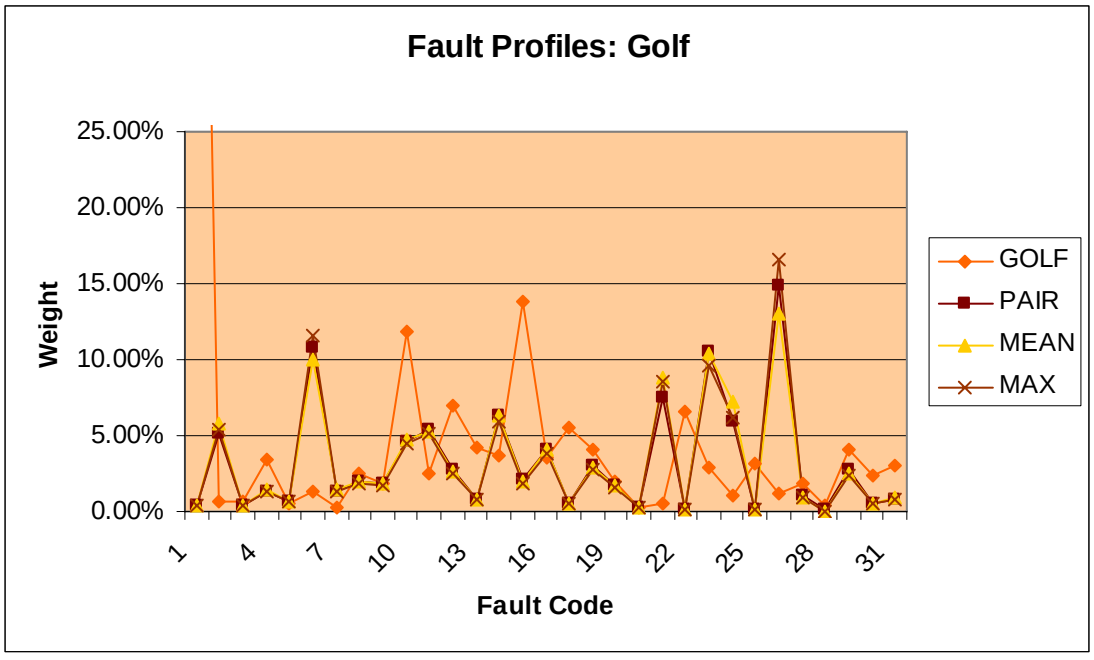



**Figure 5.3.16. Fault Profile: Golf Area**

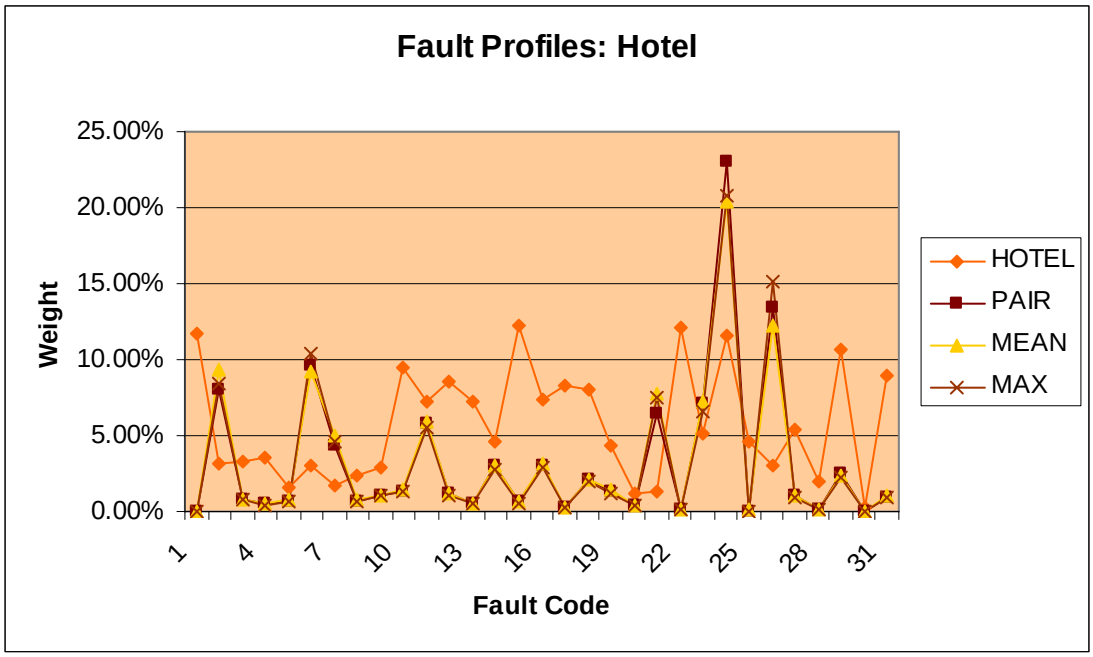



**Figure 5.3.17. Fault Profile: Hotel Area**

It is evident from the figure, that Golf and Hotel must have upward trends because the comparison profile complements entirely their existing fault profile, filling the gaps of the area profiles. This means that these two areas are quite distinct from the average systemwide profile. Faults that are common on the other areas are not common in these two. The other reason that explains the upward trend is the MED extraction, which is tightly related to the first reason.

Uncommon faults in Golf and Hotel are being enhanced by the method, while their common faults are being only moderately changed, because they are uncommon to the general group. MEDs act as a buffer for these changes. Since major events generally have high durations, they carry a lot of weight in the total results. Thus, these are a big chunk of the final variance and mean results but once removed, only the weights are determining the final result. This is the reason for the inversion of Golf area and the higher upward trend of Hotel. The major events most likely are in uncommon faults (common to Golf and Hotel though) and are not weighted heavily, which do not create a large downward change in variance. When removed though, only the weights are present and the upward trend is apparent and variance increases.

## 5.4 Other Results

As part of the initial proposal for this project, implementation and simulation of distribution networks was planned through the use of ATP. The idea was to create simulated feeder operations according to varying conditions similar to the approach on references [57] and [58]. This was to simulate normal operation, and obtain reliability values. The idea was eventually abandoned when real utility data was made available. Nevertheless, several complete feeder models were implemented before its removal from the project agenda. These models were of varying size and topologies. They were based on the IEEE Test-feeders paper [59]. All models in the paper were implemented, with some modifications for the application. Both 30 bus models (delta and wye) were implemented as well as the 13 and 120 bus feeders. The reason behind using ATP was the software's ability to solve complex systems in the time domain, and provide wave shape data; this would have been used for the evaluation of power quality and reliability data.

Another secondary area of study for this proposal was power quality considerations. Reliability and power quality generally meet in the area of short duration interruptions. Sustained interruptions are not considered for power quality since they are entirely a problem of reliability and are solved with other methods. The link between power quality and reliability is in the sub-minute to several minute (but less than 5) time frames [9]. Of course, power quality considerations are only valid when there is power; anything longer than several seconds is normally not applicable. Power quality considerations were included in the proposal for this study, and it was an objective to find suitable links between the two. Other than common pitfalls when using indices, no other relevant links were found. Also unavailability of momentary interruption data and waveshape information made it even more difficult to shed any light into potential links relevant enough to be new contribution in either power quality or reliability. However, if the appropriate data were available, further analytical connections between reliability and power quality could be explored, including the use of a similar framework as presented in this work to compare power quality performance of different regions or utilities.

# CHAPTER 6 CONCLUSIONS AND FUTURE WORK

## 6.1 Conclusions

The project had the primary objective of developing a theoretical background for the comparison of distinct distribution service areas. The development of such theory is a primary contribution of this work, along with the proposal of a mathematical process to base the comparison on. The idea behind the project was to explore the feasibility of developing a method that could equalize characteristics in a certain way, so that reliability numbers were more easily compared, evaluated and dealt with. This is critical, because the characteristics of systems make this type of comparison highly difficult, since operational bands are very different from system to system or even areas within a system. The need for universal operational bands has been increased with the advent of performance-based rates in recent years. The primary objective was achieved through the development and implementation of a new comparative framework. The framework is based on statistic theory and was validated using actual field data.

To ease the completion of the project, a modified SAIDI based index was created. The Average Normalized Daily SAIDI (ANDS) was used to compare results from different areas without having to resort to multiple yearly SAIDI plots for each area under study. The index provided results as an average and a variance, and was only one value for the entire duration of the study for each area. This decreased the analysis necessary 85% from using yearly SAIDI data for each area. The use of this index was critical for efficient study of cross-area data, and proved to be much more efficient than the use of yearly SAIDI results. ANDS is a secondary contribution of this project.

The proposed framework has proven to be a step in the right direction. It equalizes diverse areas, which could be systems if data for entire systems is available. The framework's main objective was to equalize systems with different characteristics. The American Heritage Dictionary provides this definition for equal: s*howing or having no variance in proportion, structure, or appearance* [60]. In a statistical sense, samples are equal by having similar mean value and variance. To provide equalization in this

application is to provide samples that have similar variance and respond similarly within the sample space. The results presented in Chapter 5 indicate that the developed framework indeed does this, to varying degrees. Results confirm that mean values are kept unique and not intentionally modified and variance is minimized. The objective was to strip unique characteristics which impede direct comparison.

Variance was decreased for all areas, as presented in Chapter 5, except for some areas which were special cases, and their behavior was totally justified (section 5.3.4). Variation between areas was also minimized, which would indicate that mean values were found to be closer after the application of the framework. The idea behind the process was not to artificially improve reliability results, but provide for an equal base of comparison.

From the three implemented methods for the framework, the author finds that the PAIR process is more adequate and provides better weights. This is due to the individual ordering of fault codes for each area, which is not completed by any of the other two methods (MEAN and MAXIMUM). This provides more flexibility to adapt to characteristics of the area for the comparison fault profile. Additionally it is more robust in the sense that less numerical error should occur, since $N_F$ (number of fault codes) amount of samples are used for comparison instead of one in the two other methods. In all other aspects, the three methods were similar.

Another contribution of this work is the presentation of distribution reliability indices in a more statistical sense compared to what is typically seen in the literature. By using this approach, interpretation of reliability indices and their results are made clearer. Additionally, major event considerations and their implications were discussed in detail. Finally, the critical subject of how system characteristics affect reliability was discussed in a way that is rarely discussed in available literature.

In summary, a statistical method has been implemented to create a leveled base for distribution reliability index comparison. The statistical method provides the opportunity to be implemented on systems or areas of systems and to be adaptable to the kind of faults in the systems under study. The method is also flexible enough to adapt to changing system conditions by using fault code data. The fault code data is used for

dynamic weighting of interruption data to provide adjusted results. The process is compatible with standard IEEE 1366 methods and is transparent in its execution. Major events can also be considered during the process with no conflicts. Finally it provides a more equal comparison base for dissimilar service areas.

## 6.2 *Future Work*

Future work on the subject could include a more in-depth study of fault codes in other systems and their interaction with system performance. A greater database of fault codes and their occurrence could help the assimilation of data for weighting purposes and help in the equalization or normalization of the data. This could lead to a better implementation of the weighting procedure. Additionally, a more general implementation of the framework could be achieved using fault code information from numerous systems; for example, creating a fault code database for an entire country. This could be completed by making a representative sample of utility systems across a country, and all their fault code data integrated into one database.

The framework's interaction with SAIFI could also be studied. The impact of using SAIFI values for the weighting procedures can also be evaluated and a SAIFI adjustment can also be designed. Perhaps, a hybrid SAIDI/SAIFI weighting procedure for the framework could be used, especially if detailed momentary data was made available.

With momentary data available, its impact on fault codes and performance can be studied. Momentary fault impact on reliability numbers could be analyzed and their impact on framework results. Integration into the framework could also be considered in addition to power quality considerations.

# REFERENCES


[1] IEEE, IEEE 1366-2003: IEEE Guide for Electric Power Distribution Reliability Indices. New York: Institute of Electrical and Electronics Engineers, 2003.

[2] IEEE, IEEE 1159-1995: IEEE Recommended Practice for Monitoring Electric Power Quality. New York: Institute of Electrical and Electronics Engineers, 1995.

[3] Distribution Reliability Indices Tracking within the United States, EPRI, Palo Alto, CA: 2003. 1008459.

[4] Short, T.A. Electric Power Distribution Handbook. Boca Raton, FL, USA: CRC Press, 2004.

[5] Gonën, Turan. Electric Power Distribution System Engineering. New York: McGraw-Hill, 1986.

[6] William, H.K. Distribution System Modeling and Analysis. Boca Raton, FL, USA: CRC Press. 2002.

[7] Blackburn, J.L. Protective Relaying: Principles and Applications. Second Ed. New York, NY, USA: Marcel Decker. 1998.

[8] IEEE, IEEE 493-1997: IEEE Recommended Practice for the Design of Reliable Industrial and Commercial Power Systems. New York: Institute of Electrical and Electronics Engineers, 1997.

[9] Heydt, G.T. Electric Power Quality. West LaFayette, IN, USA: Stars in a Circle,1994.

[10] Acha, Enrique and Madrigal, Manuel. Power System Harmonics: Computer Modeling and Analysis. Chichester, West Sussex, England: John Wiley & Sons, 2001.

[11] Lundquist, J., Bollen, M.H.J., "Harmonic active power flow in low and medium voltage distribution systems" *Power Engineering Society Winter Meeting, 2000*. IEEE , Volume: 4 , 2000 pp. 2858 -2863.

[12] Dugan, Roger C., McGranaghan, Mark F. and Beaty, H.W. Electric Power Systems Quality. New York, NY, USA: McGraw-Hill. 1996.

[13] Heydt, G.T. Jewell, W.T. "Pitfalls of electric power quality indices" *Power Delivery, IEEE Transactions on* , Volume: 13 Issue: 2 , Apr 1998, pp. 570 -578.

[14] Rendusara D., Von Jouanne A., Enjeti P., Paice D., "Design considerations for six pulse and twelve pulse diode rectifier systems operating under voltage unbalance and pre-existing voltage distortion with some corrective measures," *Proc. IEEE Industry Applications Conference*, Publication 95'235862, 1995, pp. 2549 - 2556.

[15] Downs, R. "Lower THD from vintage designs," *Electronics World*, v. 101, No. 106, January 1995, pp. 52-54.

[16] Shepherd, W. and Zand, P. Energy Flow and Power Factor in Nonsinusoidal Circuits, Cambridge University Press, Cambridge, UK, 1979.

[17] Xu, W. and Mansour, Y. "Developing utility harmonic regulations based on IEEE Standard 519 - BC Hydro's approach," *IEEE Transactions on Power Delivery*, v. PD-10, No. 3, July 1995, pp. 1423 – 1431.

[18] Underwriters Laboratories Standards 1561 and 1562. "Standard for dry type general purpose and power transformers," Northbrook IL, 1992.

[19] IEEE Standard C57.110-1986, IEEE recommended practice for establishing transformer capability when supplying nonsinusoidal load currents, New York, 1986.

[20] Massey, G. "Estimation method for power system harmonic effect on power distribution transformers," *IEEE Transactions on Industry Applications*, v. 30, No. 2**,** March - April, 1994, pp. 485-489.

[21] IEEE Standard 368-1977: IEEE recommended practice for measurement of electrical noise and harmonic filter performance of high-voltage direct-current systems, New York, 1997.

[22] IEEE, IEEE C37.100-1992: IEEE Standard Definitions of Power Switchgear. New York: Institute of Electrical and Electronics Engineers, 1992.

[23] IEEE, IEEE 1366-1998: IEEE Trial-Use Guide for Electric Power Distribution Reliability Indices. New York: Institute of Electrical and Electronics Engineers, 1998.

[24] Billington, R. and Pan, Z. "Incorporating Reliability Index Probability Distributions in Performance Based Regulation" *Proceedings of the 2002 IEEE Canadian Conference on Electrical and Computer Engineering.* Vol. 1, 12-15 May 2002. pp. 12-17.

[25] Huang, G.M. and Li, Y. "Power System Reliability Indices To Measure Impacts Caused by Transient Stability Crises" *Power Engineering Society Winter Meeting, 2002*. Vol. 2, 27-31 Jan. 2002. pp. 766 – 771.

[26] Knezevic, S. and Skrlec, D. "The impact of reliability indices in the process of planning radial distribution networks" *Proceedings of EUROCON 2003 Computer as a Tool*. Vol. 2, 22-24 Sept. 2003 pp. 244 – 248.

[27] Billington, R. and Bagen, "Incorporating reliability index distributions in small isolated generating system reliability performance assessment." *Generation, Transmission and Distribution, IEE Proceedings.* Vol. 151, Issue 4, 11 Jul 2004 pp. 469 – 476.

[28] Meeuwsen, J.J. and Kling, W.L. "The Influence of Protective Relay Schemes on the Reliability Indices of Load Points in Meshed Operated MV Networks."

*Conference Proceedings CIRED 97*, 2-5 June 1997, No. 438. pp. 4.14.1-4.14.5.

[29] Burke, J., "Using Outage Data to Improve Reliability." *IEEE Computer Applications in Power*. April 2000. pp. 57-60.

[30] Ayoub, A.K. and Patton, A.D., "A frequency and duration method for generating system reliability evaluation," *IEEE Transactions on Power Apparatus and Systems*, Nov. Dec. 1976, pp. 1929-1933.

[31] Billington, R., and Allan, R.N., "Reliability Evaluation of Power Systems," Plenum Publishing Corp., 1983.

[32] "A Nationwide Survey of Distribution Reliability Measurement Practices" By IEEE/PES Working Group on System Design, Paper No. 98 WM 218.

[33] Edison Electric Institute; 2000 Reliability Report; June 2001.

[34] Warren, C.M., "The Effect of Reducing Momentary Outages on Distribution Reliability Indices," *IEEE Transactions on Power Delivery*, Vol. 7, No. 3, July 1992. pp. 1610-1617.

[35] Short, T.A. EPRI PEAC Corporation, "Reliability Indices," *White Paper for T&D World*, 2002.

[36] "Five Year Electric Reliability Study 1999-2003 Oregon Investor Owned Utiltities" Prepared by the Oregon Public Utility Commission, May 2004.

[37] Warren, C.A., Ammon, R. and Welch, G., "A Survey of Distribution Reliability Measurement Practices in the U.S." *IEEE Transactions on Power Delivery*, Vol. 14, No. 1, January 1999. pp. 250-257.

[38] Warren, C.A., Pearson, D.J., and Sheehan, M.T., "A Nationwide Survey of Recorded Information Used for Calculating Distribution Reliability Indices" *IEEE Transactions on Power Delivery*, Vol. 18, No. 2, April 2003. pp. 449-453.

[39] Montgomery, Douglas C., and Runger, George C. Applied Statistics and Probability for Engineers. New York, NY, USA: John Wiley & Sons, 2003.

[40] Miller, Irwin and Freund, John E. Probability and Statistics for Engineers. Second Ed. Englewood Cliffs, NJ, USA: Prentice-Hall. 1977.

[41] Meeker, William Q., and Escobar, Luis A. Statistical Methods for Reliability Data. New York, NY, USA: John Wiley & Sons, 1998.

[42] Havil, J. Gamma: Exploring Euler's Constant. Princeton, NJ: Princeton University Press, 2003.

[43] Rothschild V. and Logothetis N. Probability Distributions. New York, NY, USA: John Wiley & Sons, 1986.

[44] Warren, C. and Saint, R. "Major Event Day calculations and how it affects small utilities" *Proceedings of the IEEE T&D Conference,* 2003. pp. D2-D2.12

[45] Warren, C. and Saint, R. "Major Event Day calculations and how it relates to small utilities" *IEEE Industry Applications Magazine*, Jan/Feb 2005. pp. 16-22.

[46] Bouford, J.D. "The Need to Segment Abnormal Events from the Calculation of Reliability Indices" Power Engineering Society Summer Meeting, 2002 IEEE Volume 2, 21-25 July 2002. pp. 636-638.

[47] Coelho, J. and Nassar, S.M. "Reliability Diagnosis of Distribution System under Adverse Weather Conditions" *Proceedings of the IEEE Bologna Powertech Conference*, June 23-26.

[48] Billington, R. and Singh, G.D. "Reliability Assessment of Transmission and Distribution Systems Considering Repair in Adverse Weather Conditions." *Proceedings of the 2002 IEEE Canadian Conference on Electrical and Computer Engineering.* Vol. 1, 12-15 May 2002. pp. 88-93.

[49] Chow, M. and Taylor, L.S. "A Novel Approach for Distribution Fault Analysis." *IEEE Transactions on Power Delivery*, Vol. 8, No. 4, October 1993. pp. 1882-1888.

[50] Chow, M. and Taylor, L.S. "Analysis and Prevention of Animal-Caused Faults in Power Distribution Systems." *IEEE Transactions on Power Delivery*, Vol. 10, No. 2, April 1995. pp. 995-1001.

[51] Lampley, G.C. "Fault detection and location on electrical distribution system case study" *Rural Electric Power Conference, 2002*. 5-7 May 2002 pp.B1 - B1_5.

[52] Burke, J.J. and Lawrence, D.J. "Characteristics of Fault Currents on Distribution Systems." *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-103, No. 1, January 1984. pp. 1-7.

[53] "Maximum Likelihood Estimates for Alpha and Beta With Zero SAIDI Days" By IEEE Working Group on System Design, February 2003.

[54] Carpaneto, E. and Chicco, G. "Evaluation of the Probability Density Functions of Distribution System Reliability Indices With a Characteristic Functions-Based Approach" *IEEE Transactions on Power Systems*. Vol. 19, Issue 2, May 2004 pp. 724 – 734.

[55] Nahman, J. and Pehric, D. "Distribution System Performance Evaluation Accounting for Data Uncertainty." *IEEE Transactions on Power Delivery*. Vol. 18, Issue 3, July 2003. pp. 694 - 700

[56] "Classification of Major Event Days" Prepared by the IEEE Working Group on System Design, Sep. 2003.

[57] Li, F. and Brown, R.E. "A Linear Contribution Factor Model of Distribution Reliability Indices and Its Applications in Monte Carlo Simulation and Sensitivity Analysis." *IEEE Transactions on Power Systems.* Vol. 18, Issue 3, Aug. 2003. pp. 1213 – 1215.

[58] Balijepalli, N. and Venkata, S.S. "Modeling and Analysis of Distribution Reliability Indices." *IEEE Transactions on Power Delivery.* Vol. 19, Issue 4, Oct. 2004. pp. 1950 – 1955.

[59] IEEE Distribution Planning Working Group Report, "Radial distribution test feeders", *IEEE Transactions on Power Systems,*, August 1991, Vol. 6, No. 3, pp. 975-985.

[60] *The American Heritage Dictionary of the English Language*, Fourth Edition. New York: Houghton Mifflin, 2000.

# APPENDICES

# APPENDIX A: RIPPER SCRIPT

```
%Matlab M file for breakingcodeappart Excel Files derived from the  dBASE
%database. It breaks appart interruption data files for each district (27)
%and makes smaller files for each feeder (1172). During the process it
%should filter bad data using the additional columns of Date and TIme check
%in the excell file. The excell file is made with the dbase file with
%several modifications to make it compatible with matlab's xlsread command.

%-------------------------------------------------------------------------%
%////////////////////////////START /////////////////////////////////////--1-%
%-------------------------------------------------------------------------%
clear all
tic
load regiones.mat
momentary=5; %Threshold value (min) for momentary events
feedernames=zeros(2,3); %Init of Feedernames. Contains feedernumber and feederclients
for each region. It is reassigned for each region iteration.
systemclients=1537306; %Value for total system clients
feedernames_size=size(feedernames); %No desc. necessary
feederoutput=zeros(feedernames_size(1,1),23);%Output file. See format in ripper.xls
feederoutputMED=zeros(feedernames_size(1,1),15);%Output file for 97 SAIDI section.
See format in ripper.xls
regdisoutput=zeros(feedernames_size(1,1),23);%Output file. Results NOT divided by
feeder clients. Used for region/district calcs.
regdisoutputMEDDED=zeros(feedernames_size(1,1),23);%Output file. MED removed.
Results NOT divided by feeder clients. Used for region/district calcs.


feederclients_size=size(feederclients);%Size of feederclients. Feederclients is loaded
from regiones.mat. It contains the master list for feeder#, clients, region and district

feederclients_append=zeros(feederclients_size(1,1),16); %Append for final feeder data
output file. Will contain data from feederoutput, that will be appended to the 4 columns
from feederclients
feederclients_append_MEDDED=zeros(feederclients_size(1,1),16);

feederindex=0; %Feedernames offset value for each region iteration

feederFAULT_append=zeros(1688,3); %Feedernumber, region, district info.
feederFAULT=zeros(1688,44); %Init for feederFAULT output files.
feederFAULT98=zeros(1688,44);
feederFAULT99=zeros(1688,44);
feederFAULT00=zeros(1688,44);
```

```
feederFAULT01=zeros(1688,44);
feederFAULT02=zeros(1688,44);
feederFAULT03=zeros(1688,44);
feederFAULT04=zeros(1688,44);
MEDcount98=0;
MEDcount99=0;
MEDcount00=0;
MEDcount01=0;
MEDcount02=0;
MEDcount03=0;
MEDcount04=0;
feederMED=zeros(1,2); %Temporary file for MED data storage
feederMEDtuki=zeros(1,10); %Stores MED thresholds for each feeder/year.

SAIDItemp=zeros(1,1);
tempsize=size(region_0);
reg0_saididaily=zeros(tempsize(1,1),10);
reg0_saididailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_1);
reg1_saididaily=zeros(tempsize(1,1),10);
reg1_saididailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_2);
reg2_saididaily=zeros(tempsize(1,1),10);
reg2_saididailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_3);
reg3_saididaily=zeros(tempsize(1,1),10);
reg3_saididailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_4);
reg4_saididaily=zeros(tempsize(1,1),10);
reg4_saididailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_5);
reg5_saididaily=zeros(tempsize(1,1),10);
reg5_saididailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_6);
reg6_saididaily=zeros(tempsize(1,1),10);
reg6_saididailyMEDDED=zeros(tempsize(1,1),10);

SAIFItemp=zeros(1,1);
tempsize=size(region_0);
reg0_saifidaily=zeros(tempsize(1,1),10);
reg0_saifidailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_1);
reg1_saifidaily=zeros(tempsize(1,1),10);
reg1_saifidailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_2);
```

```
reg2_saifidaily=zeros(tempsize(1,1),10);
reg2_saifidailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_3);
reg3_saifidaily=zeros(tempsize(1,1),10);
reg3_saifidailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_4);
reg4_saifidaily=zeros(tempsize(1,1),10);
reg4_saifidailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_5);
reg5_saifidaily=zeros(tempsize(1,1),10);
reg5_saifidailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_6);
reg6_saifidaily=zeros(tempsize(1,1),10);
reg6_saifidailyMEDDED=zeros(tempsize(1,1),10);



%-------------------------------------------------------------------------%
%////////////////////////////START OF 97 MED//////////////////////////--2-%
%-------------------------------------------------------------------------%


% ------Feeders from the known good 98-04 FIles FDTB2 are used. Later a new
% r=0:6 for will open all 7 97FDT2 files to be used for MED SAIDI
% calculations. Region 4 has no 97 season, thus no
% MED values for year 1998 will be used.
%
% All regions except #4 have 97FDT2 files. Region #4 feeders are already part of the
% feederoutput. MED process will skip region #4 calculations.

feederindex=0;
for r=0:6
   if r==0

      reg_0=region_0_97; %Stores region 97 data in temp region varible.
   end
   if r==1

      reg_0=region_1_97;
   end
   if r==2

      reg_0=region_2_97;
   end
   if r==3
```

```
    reg_0=region_3_97;
end
if r==4

    %reg_0=region_4_97

end
if r==5

    reg_0=region_5_97;
end
if r==6

    reg_0=region_6_97;
end

reg_0_size=size(reg_0);
feedernames=zeros(2,3);
feedernames_size=size(feedernames);


for i=1:reg_0_size(1,1)

    if i==1
        j=1;
        feedernames(j,1)=reg_0(i,11);
        feedernames(j,2)=1;
        feedernames(j,3)=reg_0(i,10);

    end

    if i>1 && reg_0(i,11)~=reg_0(i-1,11)
        j=j+1;

        feedernames(j,1)=reg_0(i,11);
        feedernames(j,2)=i; %Stores FIRST row where this feeder appears.
        feedernames(j,3)=reg_0(i,10);

    end
end

feedernames_size=size(feedernames);
%Create a matrix to hold the feeder data (feederoutput). One row will be created for
each
```

```
%feeder, the feeder name from variable feedernames will be included as
%column one of this matrix. A for will be used from i=1->feedernames_size,
%for filling this table with data. A nested for will rotate the reg_o
%matrix looking for data from the feeder in the i row at the time and use
%all its data to calculate SAIDI and SAIFI.

for i=1:feedernames_size(1,1)
   feederoutputMED(i+feederindex,3)=feedernames(i,1);
   feederoutputMED(i+feederindex,2)=feedernames(i,3);

   for x=1:feederclients_size(1,1)

      if feederoutputMED(i+feederindex,3)==feederclients(x,1)
         feederoutputMED(i+feederindex,1)=feederclients(x,3); %Copy Region
         feederoutputMED(i+feederindex,4)=feederclients(x,2); %Copy Clients


         if feederoutputMED(i+feederindex,1)==feederclients(x,3)
            feederoutputMED(i+feederindex,2)=feederclients(x,4);      %Copy      Right
District

         end

      end

   end
end


%----------SAIDI 97 section.

for i=1:feedernames_size(1,1)


   if i<feedernames_size(1,1)
      MEDdatalog97=zeros((feedernames(i+1,2)-feedernames(i,2)),1);

      v=1;
      for j=feedernames(i,2):(feedernames(i+1,2)-1)

         % SAIDI----------use clients(row 12) and duration (row 7)

         %feederoutput(i,1)=feedernames(i,1);
         if reg_0(j,7)>momentary %If prevents counting reclosings as part of SAIDI
            %Division by #clients will be moved inside this if
```

```
            %because: 1. Sum of ratios == a ratio of sums
            %2. THis way i can divide by the normal number of total
            %clients or those specified in the database, which
            %could be higher due to use of interties.

            if reg_0(j,12)>feederoutputMED(i+feederindex,4)
               clients=reg_0(j,12);
            else
               clients=feederoutputMED(i+feederindex,4);
            end
            if clients==0
               clients=99;
            end
            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

feederoutputMED(i+feederindex,8)=saididaily+feederoutputMED(i+feederindex,8);
%SAIDI

            if saididaily~=0

               MEDdatalog97(v,1)=log(saididaily);

               v=v+1;
            end


         end


      end
      feederoutputMED(i+feederindex,9)=mean(MEDdatalog97);
      feederoutputMED(i+feederindex,10)=std(MEDdatalog97);

feederoutputMED(i+feederindex,11)=feederoutputMED(i+feederindex,9)+2.5*feederout
putMED(i+feederindex,10);
      feederoutputMED(i+feederindex,12)=exp(feederoutputMED(i+feederindex,11));
      MEDdatalog97old=MEDdatalog97;
      MEDdatalog97old_size=size(MEDdatalog97old);
      MEDdatalog97old_append=ones(MEDdatalog97old_size(1,1),1);

MEDdatalog97old_append=MEDdatalog97old_append*feederoutputMED(i+feederindex
,3);
      MEDdatalog97old=horzcat(MEDdatalog97old_append,MEDdatalog97old);
      feederMED=vertcat(feederMED,MEDdatalog97old);
```

```
    end
    if i==feedernames_size(1,1) %If added to fix the problem when the last feeder was
read on feedernames, which had no next feeder to use as end index in the previous IF.

      v=1;
      for j=feedernames(feedernames_size(1,1),2):reg_0_size(1,1)
        MEDdatalog97=zeros(reg_0_size(1,1)-feedernames(feedernames_size(1,1),1));

        % SAIDI----------use clients(row 12) and duration (row 7)
        if reg_0(j,7)>momentary %If prevents counting reclosings as part of SAIDI

          %Division by #clients will be moved inside this if
          %because: 1. Sum of ratios == a ratio of sums
          %2. THis way i can divide by the normal number of total
          %clients or those specified in the database, which
          %could be higher due to use of interties.

          if reg_0(j,12)>feederoutputMED(i+feederindex,4)
            clients=reg_0(j,12);
          else
            clients=feederoutputMED(i+feederindex,4);
          end
          if clients==0
            clients=99;
          end
          saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

feederoutputMED(i+feederindex,8)=saididaily+feederoutputMED(i+feederindex,8);
%SAIDI

          if saididaily~=0

            MEDdatalog97(v,1)=log(saididaily);

            v=v+1;%Calculates and Stores daily SAIDI LOG
          end

          %New matrix created for storing the MEDdatalog97 for
          %each feeder, the feedernumber and the index location
          %at feederoutputMED. This index can be correlated with
          %that at feederoutput, via feedernames2.


        end
```

```
            v=v+1;

        end
        feederoutputMED(i+feederindex,9)=mean(MEDdatalog97);
        feederoutputMED(i+feederindex,10)=std(MEDdatalog97);

feederoutputMED(i+feederindex,11)=feederoutputMED(i+feederindex,9)+2.5*feederout
putMED(i+feederindex,10);
        feederoutputMED(i+feederindex,12)=exp(feederoutputMED(i+feederindex,11));
        MEDdatalog97old=MEDdatalog97;
        MEDdatalog97old_size=size(MEDdatalog97old);
        MEDdatalog97old_append=ones(MEDdatalog97old_size(1,1),1);

MEDdatalog97old_append=MEDdatalog97old_append*feederoutputMED(i+feederindex
,3);
        MEDdatalog97old=horzcat(MEDdatalog97old_append,MEDdatalog97old);
        feederMED=vertcat(feederMED,MEDdatalog97old);


      end

   end
   feederindex=feederindex+feedernames_size(1,1);
end
feederMED_size=size(feederMED);
%----FeederMED cleanup-----------------------------------------
counter=1;
for k=1:(feederMED_size)

   if feederMED(counter,2)==0
      feederMED(counter,:)=[];
      counter=counter-1;
   end
   counter=counter+1;
end




%-------------------------------------------------------------------------%
%//////////////////////////////END OF 97 CALCS//////////////////////////--2-%
%-------------------------------------------------------------------------%
```

```
%-------------------------------------------------------------------------%
%////////////////////////////START OF FEEDEROUPUT MED///////////////--3-%
%-------------------------------------------------------------------------%

feederindex=0;
for r=0:6
   if r==0
      reg_0=region_0;

   end
   if r==1
      reg_0=region_1;

   end
   if r==2
      reg_0=region_2;

   end
   if r==3
      reg_0=region_3;

   end
   if r==4
      reg_0=region_4;

   end
   if r==5
      reg_0=region_5;

   end
   if r==6
      reg_0=region_6;

   end

   reg_0_size=size(reg_0);
   feedernames=zeros(2,3);
   feedernames_size=size(feedernames);

   %---This for will read all the feeder column and extract the individual
   %feeder names, which will be later used for sorting.

   for i=1:reg_0_size(1,1)

      if i==1
```

```
        j=1;
        feedernames(j,1)=reg_0(i,11);
        feedernames(j,2)=1;
        feedernames(j,3)=reg_0(i,10);

      end
      if i>1 && reg_0(i,11)~=reg_0(i-1,11)
        j=j+1;
        feedernames(j,1)=reg_0(i,11);
        feedernames(j,2)=i; %Stores FIRST row where this feeder appears.
        feedernames(j,3)=reg_0(i,10);


      end




  end

  feedernames_size=size(feedernames);


  %Create a matrix to hold the feeder data (feederoutput). One row will be created for
each
  %feeder, the feeder name from variable feedernames will be included as
  %column one of this matrix. A for will be used from i=1->feedernames_size,
  %for filling this table with data. A nested for will rotate the reg_o
  %matrix looking for data from the feeder in the i row at the time and use
  %all its data to calculate SAIDI and SAIFI stuff.

  for i=1:feedernames_size(1,1)
    feederoutput(i+feederindex,3)=feedernames(i,1);

    for x=1:feederclients_size(1,1)

      if feederoutput(i+feederindex,3)==feederclients(x,1)
        feederoutput(i+feederindex,1)=r; %Copy R value (region raw data file index)
as region, this is used in case the feeder is non-existent in the feederclients database.
        feederoutput(i+feederindex,4)=feederclients(x,2); %Copy Clients
        feederoutput(i+feederindex,2)=feederclients(x,4); %Copy Right District
        feederoutput(i+feederindex,1)=feederclients(x,3);


        if feederoutput(i+feederindex,1)==feederclients(x,3)
```

```
            feederoutput(i+feederindex,2)=feederclients(x,4); %Copy Right District
            feederoutput(i+feederindex,4)=feederclients(x,2); %Copy Right Region


         end

      end

   end
   regdisoutput(i+feederindex,:)=feederoutput(i+feederindex,:);
end




for i=1:feedernames_size(1,1)
   a=1;
   b=1;
   c=1;
   d=1;
   e=1;
   f=1;
   g=1;
   MEDdatalog98=0;
   MEDdatalog99=0;
   MEDdatalog00=0;
   MEDdatalog01=0;
   MEDdatalog02=0;
   MEDdatalog03=0;
   MEDdatalog04=0;


   if i<feedernames_size(1,1)

      for j=feedernames(i,2):(feedernames(i+1,2)-1)

         % SAIDI----------use clients(row 12) and duration (row 7)

         %feederoutput(i,1)=feedernames(i,1);
         if reg_0(j,7)>momentary %If prevents counting reclosings as part of SAIDI
            %Division by #clients will be moved inside this if
            %because: 1. Sum of ratios == a ratio of sums
            %2. THis way i can divide by the normal number of total
            %clients or those specified in the database, which
            %could be higher due to use of interties.
```

```
            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
               if reg_0(j,12)>feederoutput(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutput(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end

               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
               feederoutput(i+feederindex,8)=saididaily+feederoutput(i+feederindex,8);
%SAIDI
               saididailynumonly=(reg_0(j,12)*reg_0(j,7));

regdisoutput(i+feederindex,8)=saididailynumonly+regdisoutput(i+feederindex,8);

               if saididaily~=0

                  MEDdatalog98(a,1)=log(saididaily);

                  a=a+1;
               end
            end
            if reg_0(j,1)>=36161 && reg_0(j,1)<=36525
               if reg_0(j,12)>feederoutput(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutput(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end

               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
               feederoutput(i+feederindex,9)=saididaily+feederoutput(i+feederindex,9);
%SAIDI
               saididailynumonly=(reg_0(j,12)*reg_0(j,7));

regdisoutput(i+feederindex,9)=saididailynumonly+regdisoutput(i+feederindex,9);

               if saididaily~=0
```

```
                MEDdatalog99(b,1)=log(saididaily);

                b=b+1;
            end
        end
        if reg_0(j,1)>=36526 && reg_0(j,1)<=36891
            if reg_0(j,12)>feederoutput(i+feederindex,4)
                clients=reg_0(j,12);
            else
                clients=feederoutput(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end

            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

feederoutput(i+feederindex,10)=saididaily+feederoutput(i+feederindex,10); %SAIDI
            saididailynumonly=(reg_0(j,12)*reg_0(j,7));

regdisoutput(i+feederindex,10)=saididailynumonly+regdisoutput(i+feederindex,10);

            if saididaily~=0

                MEDdatalog00(c,1)=log(saididaily);

                c=c+1;
            end
        end
        if reg_0(j,1)>=36892 && reg_0(j,1)<=37256
            if reg_0(j,12)>feederoutput(i+feederindex,4)
                clients=reg_0(j,12);
            else
                clients=feederoutput(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end

            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

feederoutput(i+feederindex,11)=saididaily+feederoutput(i+feederindex,11); %SAIDI
            saididailynumonly=(reg_0(j,12)*reg_0(j,7));
```

```
regdisoutput(i+feederindex,11)=saididailynumonly+regdisoutput(i+feederindex,11);

                if saididaily~=0

                    MEDdatalog01(d,1)=log(saididaily);

                    d=d+1;
                end
            end
            if reg_0(j,1)>=37257 && reg_0(j,1)<=37621
                if reg_0(j,12)>feederoutput(i+feederindex,4)
                    clients=reg_0(j,12);
                else
                    clients=feederoutput(i+feederindex,4);
                end
                if clients==0
                    clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

feederoutput(i+feederindex,12)=saididaily+feederoutput(i+feederindex,12); %SAIDI
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

regdisoutput(i+feederindex,12)=saididailynumonly+regdisoutput(i+feederindex,12);

                if saididaily~=0

                    MEDdatalog02(e,1)=log(saididaily);

                    e=e+1;
                end
            end
            if reg_0(j,1)>=37622 && reg_0(j,1)<=37986
                if reg_0(j,12)>feederoutput(i+feederindex,4)
                    clients=reg_0(j,12);
                else
                    clients=feederoutput(i+feederindex,4);
                end
                if clients==0
                    clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
```

```
feederoutput(i+feederindex,13)=saididaily+feederoutput(i+feederindex,13); %SAIDI
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

regdisoutput(i+feederindex,13)=saididailynumonly+regdisoutput(i+feederindex,13);

                if saididaily~=0

                   MEDdatalog03(f,1)=log(saididaily);

                   f=f+1;
                end
             end
             if reg_0(j,1)>=37987 && reg_0(j,1)<=38352
                if reg_0(j,12)>feederoutput(i+feederindex,4)
                   clients=reg_0(j,12);
                else
                   clients=feederoutput(i+feederindex,4);
                end
                if clients==0
                   clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

feederoutput(i+feederindex,14)=saididaily+feederoutput(i+feederindex,14); %SAIDI
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

regdisoutput(i+feederindex,14)=saididailynumonly+regdisoutput(i+feederindex,14);

                if saididaily~=0

                   MEDdatalog04(g,1)=log(saididaily);

                   g=g+1;
                end
             end




          end

       end
       %New MED module.....lets see if it works. :o
```

```
%Need to chk feeder number for each iteration, search its
%97MEDlog data, and copy it to the new log file.Thus, i need to
%read from feederoutput, and read in a for feederMED.
feederMED_size=size(feederMED);
MEDdatalogA=0;
t=1;
for u=1:feederMED_size(1,1)
   if feederMED(u,1)==feederoutput(i+feederindex,3)
      MEDdatalogA(t,1)=feederMED(u,2);
      t=t+1;
   end
end




MEDdatalogB=vertcat(MEDdatalogA,MEDdatalog98);
MEDdatalogC=vertcat(MEDdatalogB,MEDdatalog99);
MEDdatalogD=vertcat(MEDdatalogC,MEDdatalog00);
MEDdatalogE=vertcat(MEDdatalogD,MEDdatalog01);

MEDdatalogF=vertcat(MEDdatalog98,MEDdatalog99,MEDdatalog00,MEDdatalog01,M
EDdatalog02);

MEDdatalogG=vertcat(MEDdatalog99,MEDdatalog00,MEDdatalog01,MEDdatalog02,
MEDdatalog03);



feederMEDtuki(i+feederindex,1)=feederoutput(i+feederindex,1);
feederMEDtuki(i+feederindex,2)=feederoutput(i+feederindex,2);
feederMEDtuki(i+feederindex,3)=feederoutput(i+feederindex,3);
meantempA=mean(MEDdatalogA);
devtempA=std(MEDdatalogA);
combotempA=meantempA+2.5*devtempA;
feederMEDtuki(i+feederindex,4)=exp(combotempA);

meantempB=mean(MEDdatalogB);
devtempB=std(MEDdatalogB);
combotempB=meantempB+2.5*devtempB;
feederMEDtuki(i+feederindex,5)=exp(combotempB);

meantempC=mean(MEDdatalogC);
devtempC=std(MEDdatalogC);
combotempC=meantempC+2.5*devtempC;
```

```
    feederMEDtuki(i+feederindex,6)=exp(combotempC);

    meantempD=mean(MEDdatalogD);
    devtempD=std(MEDdatalogD);
    combotempD=meantempD+2.5*devtempD;
    feederMEDtuki(i+feederindex,7)=exp(combotempD);

    meantempE=mean(MEDdatalogE);
    devtempE=std(MEDdatalogE);
    combotempE=meantempE+2.5*devtempE;
    feederMEDtuki(i+feederindex,8)=exp(combotempE);

    meantempF=mean(MEDdatalogF);
    devtempF=std(MEDdatalogF);
    combotempF=meantempF+2.5*devtempF;
    feederMEDtuki(i+feederindex,9)=exp(combotempF);

    meantempG=mean(MEDdatalogG);
    devtempG=std(MEDdatalogG);
    combotempG=meantempG+2.5*devtempG;
    feederMEDtuki(i+feederindex,10)=exp(combotempG);

end
```

if i==feedernames_size(1,1) %If added to fix the problem when the last feeder was read on feedernames, which had no next feeder to use as end index in the previous IF.

```
    for j=feedernames(feedernames_size(1,1),2):reg_0_size(1,1)

        % SAIDI----------use clients(row 12) and duration (row 7)
        if reg_0(j,7)>momentary %If prevents counting reclosings as part of SAIDI

            %Division by #clients will be moved inside this if
            %because: 1. Sum of ratios == a ratio of sums
            %2. THis way i can divide by the normal number of total
            %clients or those specified in the database, which
            %could be higher due to use of interties.



            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
                if reg_0(j,12)>feederoutput(i+feederindex,4)
                    clients=reg_0(j,12);
```

```
                else
                    clients=feederoutput(i+feederindex,4);
                end
                if clients==0
                    clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
                feederoutput(i+feederindex,8)=saididaily+feederoutput(i+feederindex,8);
%SAIDI
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

regdisoutput(i+feederindex,8)=saididailynumonly+regdisoutput(i+feederindex,8);

                if saididaily~=0

                    MEDdatalog98(a,1)=log(saididaily);

                    a=a+1;
                end
            end
            if reg_0(j,1)>=36161 && reg_0(j,1)<=36525
                if reg_0(j,12)>feederoutput(i+feederindex,4)
                    clients=reg_0(j,12);
                else
                    clients=feederoutput(i+feederindex,4);
                end
                if clients==0
                    clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
                feederoutput(i+feederindex,9)=saididaily+feederoutput(i+feederindex,9);
%SAIDI
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

regdisoutput(i+feederindex,9)=saididailynumonly+regdisoutput(i+feederindex,9);

                if saididaily~=0

                    MEDdatalog99(b,1)=log(saididaily);

                    b=b+1;
                end
            end
```

```
            if reg_0(j,1)>=36526 && reg_0(j,1)<=36891
               if reg_0(j,12)>feederoutput(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutput(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end

               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

feederoutput(i+feederindex,10)=saididaily+feederoutput(i+feederindex,10); %SAIDI
               saididailynumonly=(reg_0(j,12)*reg_0(j,7));

regdisoutput(i+feederindex,10)=saididailynumonly+regdisoutput(i+feederindex,10);

               if saididaily~=0

                  MEDdatalog00(c,1)=log(saididaily);

                  c=c+1;
               end
            end
            if reg_0(j,1)>=36892 && reg_0(j,1)<=37256
               if reg_0(j,12)>feederoutput(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutput(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end

               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

feederoutput(i+feederindex,11)=saididaily+feederoutput(i+feederindex,11); %SAIDI
               saididailynumonly=(reg_0(j,12)*reg_0(j,7));

regdisoutput(i+feederindex,11)=saididailynumonly+regdisoutput(i+feederindex,11);

               if saididaily~=0

                  MEDdatalog01(d,1)=log(saididaily);
```

```
            d=d+1;
        end
    end
    if reg_0(j,1)>=37257 && reg_0(j,1)<=37621
        if reg_0(j,12)>feederoutput(i+feederindex,4)
            clients=reg_0(j,12);
        else
            clients=feederoutput(i+feederindex,4);
        end
        if clients==0
            clients=99;
        end

        saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

feederoutput(i+feederindex,12)=saididaily+feederoutput(i+feederindex,12); %SAIDI
        saididailynumonly=(reg_0(j,12)*reg_0(j,7));

regdisoutput(i+feederindex,12)=saididailynumonly+regdisoutput(i+feederindex,12);

        if saididaily~=0

            MEDdatalog02(e,1)=log(saididaily);

            e=e+1;
        end
    end
    if reg_0(j,1)>=37622 && reg_0(j,1)<=37986
        if reg_0(j,12)>feederoutput(i+feederindex,4)
            clients=reg_0(j,12);
        else
            clients=feederoutput(i+feederindex,4);
        end
        if clients==0
            clients=99;
        end

        saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

feederoutput(i+feederindex,13)=saididaily+feederoutput(i+feederindex,13); %SAIDI
        saididailynumonly=(reg_0(j,12)*reg_0(j,7));

regdisoutput(i+feederindex,13)=saididailynumonly+regdisoutput(i+feederindex,13);

        if saididaily~=0
```

```
                MEDdatalog03(f,1)=log(saididaily);

                f=f+1;
              end
            end
            if reg_0(j,1)>=37987 && reg_0(j,1)<=38352
              if reg_0(j,12)>feederoutput(i+feederindex,4)
                clients=reg_0(j,12);
              else
                clients=feederoutput(i+feederindex,4);
              end
              if clients==0
                clients=99;
              end

              saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

feederoutput(i+feederindex,14)=saididaily+feederoutput(i+feederindex,14); %SAIDI
              saididailynumonly=(reg_0(j,12)*reg_0(j,7));

regdisoutput(i+feederindex,14)=saididailynumonly+regdisoutput(i+feederindex,14);

              if saididaily~=0

                MEDdatalog04(g,1)=log(saididaily);

                g=g+1;
              end
            end
          end

        end
        MEDdatalogA=0;
        t=1;
        for u=1:feederMED_size(1,1)
          if feederMED(u,1)==feederoutput(i+feederindex,3)
            MEDdatalogA(t,1)=feederMED(u,2);
            t=t+1;
          end
        end


        MEDdatalogB=vertcat(MEDdatalogA,MEDdatalog98);
        MEDdatalogC=vertcat(MEDdatalogB,MEDdatalog99);
```

```
        MEDdatalogD=vertcat(MEDdatalogC,MEDdatalog00);
        MEDdatalogE=vertcat(MEDdatalogD,MEDdatalog01);

MEDdatalogF=vertcat(MEDdatalog98,MEDdatalog99,MEDdatalog00,MEDdatalog01,M
EDdatalog02);

MEDdatalogG=vertcat(MEDdatalog99,MEDdatalog00,MEDdatalog01,MEDdatalog02,
MEDdatalog03);

        feederMEDtuki(i+feederindex,1)=feederoutput(i+feederindex,1);
        feederMEDtuki(i+feederindex,2)=feederoutput(i+feederindex,2);
        feederMEDtuki(i+feederindex,3)=feederoutput(i+feederindex,3);
        meantempA=mean(MEDdatalogA);
        devtempA=std(MEDdatalogA);
        combotempA=meantempA+2.5*devtempA;
        feederMEDtuki(i+feederindex,4)=exp(combotempA);

        meantempB=mean(MEDdatalogB);
        devtempB=std(MEDdatalogB);
        combotempB=meantempB+2.5*devtempB;
        feederMEDtuki(i+feederindex,5)=exp(combotempB);

        meantempC=mean(MEDdatalogC);
        devtempC=std(MEDdatalogC);
        combotempC=meantempC+2.5*devtempC;
        feederMEDtuki(i+feederindex,6)=exp(combotempC);

        meantempD=mean(MEDdatalogD);
        devtempD=std(MEDdatalogD);
        combotempD=meantempD+2.5*devtempD;
        feederMEDtuki(i+feederindex,7)=exp(combotempD);

        meantempE=mean(MEDdatalogE);
        devtempE=std(MEDdatalogE);
        combotempE=meantempE+2.5*devtempE;
        feederMEDtuki(i+feederindex,8)=exp(combotempE);

        meantempF=mean(MEDdatalogF);
        devtempF=std(MEDdatalogF);
        combotempF=meantempF+2.5*devtempF;
        feederMEDtuki(i+feederindex,9)=exp(combotempF);

        meantempG=mean(MEDdatalogG);
        devtempG=std(MEDdatalogG);
        combotempG=meantempG+2.5*devtempG;
```

```
        feederMEDtuki(i+feederindex,10)=exp(combotempG);



      end

      if feederoutput(i+feederindex,4)==0
        feederoutput(i+feederindex,8)=999;
        feederoutput(i+feederindex,9)=999;
        feederoutput(i+feederindex,10)=999;
        feederoutput(i+feederindex,11)=999;
        feederoutput(i+feederindex,12)=999;
        feederoutput(i+feederindex,13)=999;
        feederoutput(i+feederindex,14)=999;
      end
    end


    feederoutput_size=size(feederoutput);

    feederindex=feederindex+feedernames_size(1,1);

end
%-------------------------------------------------------------------------%
%////////////////////////////END OF FEEDEROUPUT MED/////////////////--3-%
%-------------------------------------------------------------------------%


%-------------------------------------------------------------------------%
%////////////////////////////ALL MEDDED SAIDI SAIFI/////////////////--4-%
%-------------------------------------------------------------------------%
feederoutputMEDDED=zeros(feederoutput_size);
feederindex=0;
for r=0:6
    if r==0
        reg_0=region_0;
        regionmarker(1,1)=1;
        saididailytemp=reg0_saididaily;
        saididailytempMEDDED=reg0_saididailyMEDDED;
        saifidailytemp=reg0_saifidaily;
        saifidailytempMEDDED=reg0_saifidailyMEDDED
    end
    if r==1
        reg_0=region_1;
        regionmarker(2,1)=feederindex;
```

```
    saididailytemp=reg1_saididaily;
    saididailytempMEDDED=reg1_saididailyMEDDED;
    saifidailytemp=reg1_saifidaily;
    saifidailytempMEDDED=reg1_saifidailyMEDDED;
end
if r==2
    reg_0=region_2;
    regionmarker(3,1)=feederindex;
    saididailytemp=reg2_saididaily;
    saididailytempMEDDED=reg2_saididailyMEDDED;
    saididailytemp=reg2_saifidaily;
    saididailytempMEDDED=reg2_saifidailyMEDDED;
end
if r==3
    reg_0=region_3;
    regionmarker(4,1)=feederindex;
    saididailytemp=reg3_saididaily;
    saididailytempMEDDED=reg3_saididailyMEDDED;
    saididailytemp=reg3_saifidaily;
    saididailytempMEDDED=reg3_saifidailyMEDDED;
end
if r==4
    reg_0=region_4;
    regionmarker(5,1)=feederindex;
    saididailytemp=reg4_saididaily;
    saididailytempMEDDED=reg4_saididailyMEDDED;
    saifidailytemp=reg4_saifidaily;
    saifidailytempMEDDED=reg4_saifidailyMEDDED;
end
if r==5
    reg_0=region_5;
    regionmarker(6,1)=feederindex;
    saididailytemp=reg5_saididaily;
    saididailytempMEDDED=reg5_saifidailyMEDDED;
    saifidailytemp=reg5_saididaily;
    saifidailytempMEDDED=reg5_saifidailyMEDDED;
end
if r==6
    reg_0=region_6;
    regionmarker(7,1)=feederindex;
    saididailytemp=reg6_saididaily;
    saididailytempMEDDED=reg6_saididailyMEDDED;
    saifidailytemp=reg6_saifidaily;
    saifidailytempMEDDED=reg6_saifidailyMEDDED;
end
```

```
reg_0_size=size(reg_0);
feedernames=zeros(2,3);
feedernames_size=size(feedernames);

%---This for will read all the feeder column and extract the individual
%feeder names, which will be later used for sorting.

for i=1:reg_0_size(1,1)

   if i==1
      j=1;
      feedernames(j,1)=reg_0(i,11);
      feedernames(j,2)=1;
      feedernames(j,3)=reg_0(i,10);


   end
   if i>1 && reg_0(i,11)~=reg_0(i-1,11)
      j=j+1;

      feedernames(j,1)=reg_0(i,11);
      feedernames(j,2)=i; %Stores FIRST row where this feeder appears in XLS file.
      feedernames(j,3)=reg_0(i,10);

   end



end

feedernames_size=size(feedernames);
%Create a matrix to hold the feeder data (feederoutput). One row will be created for
each
%feeder, the feeder name from variable feedernames will be included as
%column one of this matrix. A for will be used from i=1->feedernames_size,
%for filling this table with data. A nested for will rotate the reg_o
%matrix looking for data from the feeder in the i row at the time and use
%all its data to calculate SAIDI and SAIFI stuff.

for i=1:feedernames_size(1,1)
   feederoutputMEDDED(i+feederindex,3)=feedernames(i,1);
   %feederoutputMEDDED(i+feederindex,2)=feedernames(i,3);

   for x=1:feederclients_size(1,1)
```

```
        if feederoutputMEDDED(i+feederindex,3)==feederclients(x,1)
            feederoutputMEDDED(i+feederindex,1)=r; %Copy R value (region raw data file index) as region, this is used in case the feeder is non-existent in the feederclients database.
            feederoutputMEDDED(i+feederindex,4)=feederclients(x,2); %Copy Clients
            feederoutputMEDDED(i+feederindex,2)=feederclients(x,4);    %Copy    Right District
            feederoutputMEDDED(i+feederindex,1)=feederclients(x,3);


            if feederoutputMEDDED(i+feederindex,1)==feederclients(x,3)
               feederoutputMEDDED(i+feederindex,2)=feederclients(x,4);   %Copy   Right District
               feederoutputMEDDED(i+feederindex,4)=feederclients(x,2);   %Copy   Right Region


            end

         end

      end
      regdisoutputMEDDED(i+feederindex,:)=feederoutputMEDDED(i+feederindex,:);
   end


%*********************************************************************
***
   %-----SAIDI W/O MED + FAULT CODE EXTRACTION----------------

%*********************************************************************
***


   for i=1:feedernames_size(1,1)
      feederFAULT_append(i+feederindex,1)=feederoutput(i+feederindex,1);
      feederFAULT_append(i+feederindex,2)=feederoutput(i+feederindex,2);
      feederFAULT_append(i+feederindex,3)=feederoutput(i+feederindex,3);

      if i<feedernames_size(1,1)

         for j=feedernames(i,2):(feedernames(i+1,2)-1)

            % SAIDI----------use clients(row 12) and duration (row 7)
```

```
%feederoutputMEDDED(i,1)=feedernames(i,1);
if reg_0(j,7)>momentary %If prevents counting reclosings as part of SAIDI
   %Division by #clients will be moved inside this if
   %because: 1. Sum of ratios == a ratio of sums
   %2. THis way i can divide by the normal number of total
   %clients or those specified in the database, which
   %could be higher due to use of interties.


   if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
      if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
         clients=reg_0(j,12);
      else
         clients=feederoutputMEDDED(i+feederindex,4);
      end
      if clients==0
         clients=99;
      end
      saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
      saididailynumonly=(reg_0(j,12)*reg_0(j,7));
      if clients>0
         saididailytemp(j,1)=reg_0(j,11);
         saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
         saididailytemp(j,3)=clients;
         saididailytemp(j,4)=saididailynumonly;
      end
      if (saididaily>=feederMEDtuki(i+feederindex,4))&&(r~=4)
         saididaily=0;
         saididailynumonly=0;
         MEDcount98=MEDcount98+1;
      end
      if clients<=50
         saididaily=0;
         saididailynumonly=0;
      end
      if reg_0(j,13)==38 || reg_0(j,13)==39 %Elimina operaciones en el 38 o
115
         saididaily=0;
         saididailynumonly=0;

      end

feederoutputMEDDED(i+feederindex,8)=saididaily+feederoutputMEDDED(i+feederind
ex,8); %SAIDI
```

```
regdisoutputMEDDED(i+feederindex,8)=saididailynumonly+regdisoutputMEDDED(i+f
eederindex,8);
                    if clients>0
                       saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                       saididailytempMEDDED(j,3)=clients;
                       saididailytempMEDDED(j,4)=saididailynumonly;
                    end


                end
                if reg_0(j,1)>=36161 && reg_0(j,1)<=36525
                   if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                      clients=reg_0(j,12);
                   else
                      clients=feederoutputMEDDED(i+feederindex,4);
                   end
                   if clients==0
                      clients=99;
                   end

                   saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
                   saididailynumonly=(reg_0(j,12)*reg_0(j,7));
                   if clients>0
                      saididailytemp(j,1)=reg_0(j,11);
                      saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                      saididailytemp(j,3)=clients;
                      saididailytemp(j,5)=saididailynumonly;
                   end
                   if saididaily>=feederMEDtuki(i+feederindex,5)
                      saididaily=0;
                      saididailynumonly=0;
                      MEDcount99=MEDcount99+1;
                   end
                   if clients<=50
                      saididaily=0;
                      saididailynumonly=0;
                   end
                   if reg_0(j,13)==38 || reg_0(j,13)==39
                      saididaily=0;
                      saididailynumonly=0;
                   end
```

```
feederoutputMEDDED(i+feederindex,9)=saididaily+feederoutputMEDDED(i+feederind
ex,9); %SAIDI

regdisoutputMEDDED(i+feederindex,9)=saididailynumonly+regdisoutputMEDDED(i+f
eederindex,9);
                    if clients>0
                       saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                       saididailytempMEDDED(j,3)=clients;
                       saididailytempMEDDED(j,5)=saididailynumonly;
                    end


                end
                if reg_0(j,1)>=36526 && reg_0(j,1)<=36891
                   if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                      clients=reg_0(j,12);
                   else
                      clients=feederoutputMEDDED(i+feederindex,4);
                   end
                   if clients==0
                      clients=99;
                   end

                   saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
                   saididailynumonly=(reg_0(j,12)*reg_0(j,7));
                   if clients>0
                      saididailytemp(j,1)=reg_0(j,11);
                      saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                      saididailytemp(j,3)=clients;
                      saididailytemp(j,6)=saididailynumonly;
                   end
                   if saididaily>=feederMEDtuki(i+feederindex,6)
                      saididaily=0;
                      saididailynumonly=0;
                      MEDcount00=MEDcount00+1;
                   end
                   if clients<=50
                      saididaily=0;
                      saididailynumonly=0;
                   end
                   if reg_0(j,13)==38 || reg_0(j,13)==39
                      saididaily=0;
```

```
                saididailynumonly=0;

            end

feederoutputMEDDED(i+feederindex,10)=saididaily+feederoutputMEDDED(i+feederin
dex,10); %SAIDI

regdisoutputMEDDED(i+feederindex,10)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,10);
            if clients>0
                saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytempMEDDED(j,3)=clients;
                saididailytempMEDDED(j,6)=saididailynumonly;
            end

        end
        if reg_0(j,1)>=36892 && reg_0(j,1)<=37256
            if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                clients=reg_0(j,12);
            else
                clients=feederoutputMEDDED(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end

            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
            saididailynumonly=(reg_0(j,12)*reg_0(j,7));
            if clients>0
                saididailytemp(j,1)=reg_0(j,11);
                saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytemp(j,3)=clients;
                saididailytemp(j,7)=saididailynumonly;
            end
            if saididaily>=feederMEDtuki(i+feederindex,7)
                saididaily=0;
                saididailynumonly=0;
                MEDcount01=MEDcount01+1;
            end
            if clients<=50
                saididaily=0;
                saididailynumonly=0;
            end
```

```
                if reg_0(j,13)==38 || reg_0(j,13)==39
                    saididaily=0;
                    saididailynumonly=0;

                end

feederoutputMEDDED(i+feederindex,11)=saididaily+feederoutputMEDDED(i+feederin
dex,11); %SAIDI

regdisoutputMEDDED(i+feederindex,11)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,11);
                if clients>0
                    saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                    saididailytempMEDDED(j,3)=clients;
                    saididailytempMEDDED(j,7)=saididailynumonly;
                end

            end
            if reg_0(j,1)>=37257 && reg_0(j,1)<=37621
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                    clients=reg_0(j,12);
                else
                    clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                    clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));
                if clients>0
                    saididailytemp(j,1)=reg_0(j,11);
                    saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                    saididailytemp(j,3)=clients;
                    saididailytemp(j,8)=saididailynumonly;
                end
                if saididaily>=feederMEDtuki(i+feederindex,8)
                    saididaily=0;
                    saididailynumonly=0;
                    MEDcount02=MEDcount02+1;
                end
                if clients<=50
                    saididaily=0;
```

```
                saididailynumonly=0;
            end
            if reg_0(j,13)==38 || reg_0(j,13)==39
                saididaily=0;
                saididailynumonly=0;

            end

feederoutputMEDDED(i+feederindex,12)=saididaily+feederoutputMEDDED(i+feederin
dex,12); %SAIDI

regdisoutputMEDDED(i+feederindex,12)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,12);
            if clients>0
                saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytempMEDDED(j,3)=clients;
                saididailytempMEDDED(j,8)=saididailynumonly;
            end

        end
        if reg_0(j,1)>=37622 && reg_0(j,1)<=37986
            if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                clients=reg_0(j,12);
            else
                clients=feederoutputMEDDED(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end

            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
            saididailynumonly=(reg_0(j,12)*reg_0(j,7));
            if clients>0
                saididailytemp(j,1)=reg_0(j,11);
                saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytemp(j,3)=clients;
                saididailytemp(j,9)=saididailynumonly;
            end
            if saididaily>=feederMEDtuki(i+feederindex,9)
                saididaily=0;
                saididailynumonly=0;
                MEDcount03=MEDcount03+1;
            end
```

```
                if clients<=50
                   saididaily=0;
                   saididailynumonly=0;
                end
                if reg_0(j,13)==38 || reg_0(j,13)==39
                   saididaily=0;
                   saididailynumonly=0;

                end

feederoutputMEDDED(i+feederindex,13)=saididaily+feederoutputMEDDED(i+feederin
dex,13); %SAIDI

regdisoutputMEDDED(i+feederindex,13)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,13);
                if clients>0
                   saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytempMEDDED(j,3)=clients;
                   saididailytempMEDDED(j,9)=saididailynumonly;
                end


             end
             if reg_0(j,1)>=37987 && reg_0(j,1)<=38352
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                   clients=reg_0(j,12);
                else
                   clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                   clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));
                if clients>0
                   saididailytemp(j,1)=reg_0(j,11);
                   saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytemp(j,3)=clients;
                   saididailytemp(j,10)=saididailynumonly;
                end
                if saididaily>=feederMEDtuki(i+feederindex,10)
                   saididaily=0;
```

```
                saididailynumonly=0;
                MEDcount04=MEDcount04+1;
            end
            if clients<=50
                saididaily=0;
                saididailynumonly=0;
            end
            if reg_0(j,13)==38 || reg_0(j,13)==39
                saididaily=0;
                saididailynumonly=0;

            end

feederoutputMEDDED(i+feederindex,14)=saididaily+feederoutputMEDDED(i+feederin
dex,14); %SAIDI

regdisoutputMEDDED(i+feederindex,14)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,14);
            if clients>0
                saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytempMEDDED(j,3)=clients;
                saididailytempMEDDED(j,10)=saididailynumonly;
            end

        end

    end
    %FAULT EXTRACTOR V2

    if reg_0(j,13)==13
        if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
            feederFAULT98(i+feederindex,4)=feederFAULT98(i+feederindex,4)+1;
        elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525
            feederFAULT99(i+feederindex,4)=feederFAULT99(i+feederindex,4)+1;
        elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891
            feederFAULT00(i+feederindex,4)=feederFAULT00(i+feederindex,4)+1;
        elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256
            feederFAULT01(i+feederindex,4)=feederFAULT01(i+feederindex,4)+1;
        elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621
            feederFAULT02(i+feederindex,4)=feederFAULT02(i+feederindex,4)+1;
        elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986
            feederFAULT03(i+feederindex,4)=feederFAULT03(i+feederindex,4)+1;
        elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352
```

```
        feederFAULT04(i+feederindex,4)=feederFAULT04(i+feederindex,4)+1;
    end

    feederFAULT(i+feederindex,4)=feederFAULT(i+feederindex,4)+1;
end
if reg_0(j,13)==38
    if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

        feederFAULT98(i+feederindex,5)=feederFAULT98(i+feederindex,5)+1;
    elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

        feederFAULT99(i+feederindex,5)=feederFAULT99(i+feederindex,5)+1;
    elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

        feederFAULT00(i+feederindex,5)=feederFAULT00(i+feederindex,5)+1;
    elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

        feederFAULT01(i+feederindex,5)=feederFAULT01(i+feederindex,5)+1;
    elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

        feederFAULT02(i+feederindex,5)=feederFAULT02(i+feederindex,5)+1;
    elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

        feederFAULT03(i+feederindex,5)=feederFAULT03(i+feederindex,5)+1;
    elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

        feederFAULT04(i+feederindex,5)=feederFAULT04(i+feederindex,5)+1;
    end
    feederFAULT(i+feederindex,5)=feederFAULT(i+feederindex,5)+1;
end
if reg_0(j,13)==39

    if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
        feederFAULT98(i+feederindex,6)=feederFAULT98(i+feederindex,6)+1;
    elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525
        feederFAULT99(i+feederindex,6)=feederFAULT99(i+feederindex,6)+1;
    elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891
        feederFAULT00(i+feederindex,6)=feederFAULT00(i+feederindex,6)+1;
    elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256
        feederFAULT01(i+feederindex,6)=feederFAULT01(i+feederindex,6)+1;
    elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621
        feederFAULT02(i+feederindex,6)=feederFAULT02(i+feederindex,6)+1;
    elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986
        feederFAULT03(i+feederindex,6)=feederFAULT03(i+feederindex,6)+1;
    elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352
```

```
      feederFAULT04(i+feederindex,6)=feederFAULT04(i+feederindex,6)+1;
   end
   feederFAULT(i+feederindex,6)=feederFAULT(i+feederindex,6)+1;
end
if reg_0(j,13)==48

   if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

      feederFAULT98(i+feederindex,7)=feederFAULT98(i+feederindex,7)+1;
   elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

      feederFAULT99(i+feederindex,7)=feederFAULT99(i+feederindex,7)+1;
   elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

      feederFAULT00(i+feederindex,7)=feederFAULT00(i+feederindex,7)+1;
   elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

      feederFAULT01(i+feederindex,7)=feederFAULT01(i+feederindex,7)+1;
   elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

      feederFAULT02(i+feederindex,7)=feederFAULT02(i+feederindex,7)+1;
   elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

      feederFAULT03(i+feederindex,7)=feederFAULT03(i+feederindex,7)+1;
   elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

      feederFAULT04(i+feederindex,7)=feederFAULT04(i+feederindex,7)+1;
   end
   feederFAULT(i+feederindex,7)=feederFAULT(i+feederindex,7)+1;
end
if reg_0(j,13)==51

   if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
      feederFAULT98(i+feederindex,8)=feederFAULT98(i+feederindex,8)+1;
   elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525
      feederFAULT99(i+feederindex,8)=feederFAULT99(i+feederindex,8)+1;
   elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891
      feederFAULT00(i+feederindex,8)=feederFAULT00(i+feederindex,8)+1;
   elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256
      feederFAULT01(i+feederindex,8)=feederFAULT01(i+feederindex,8)+1;
   elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621
      feederFAULT02(i+feederindex,8)=feederFAULT02(i+feederindex,8)+1;
   elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986
      feederFAULT03(i+feederindex,8)=feederFAULT03(i+feederindex,8)+1;
   elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352
```

```
            feederFAULT04(i+feederindex,8)=feederFAULT04(i+feederindex,8)+1;
          end
          feederFAULT(i+feederindex,8)=feederFAULT(i+feederindex,8)+1;
        end
        if reg_0(j,13)==52

          if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
            feederFAULT98(i+feederindex,9)=feederFAULT98(i+feederindex,9)+1;
          elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525
            feederFAULT99(i+feederindex,9)=feederFAULT99(i+feederindex,9)+1;
          elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891
            feederFAULT00(i+feederindex,9)=feederFAULT00(i+feederindex,9)+1;
          elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256
            feederFAULT01(i+feederindex,9)=feederFAULT01(i+feederindex,9)+1;
          elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621
            feederFAULT02(i+feederindex,9)=feederFAULT02(i+feederindex,9)+1;
          elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986
            feederFAULT03(i+feederindex,9)=feederFAULT03(i+feederindex,9)+1;
          elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352
            feederFAULT04(i+feederindex,9)=feederFAULT04(i+feederindex,9)+1;
          end
          feederFAULT(i+feederindex,9)=feederFAULT(i+feederindex,9)+1;
        end
        if reg_0(j,13)==53

          if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,10)=feederFAULT98(i+feederindex,10)+1;
          elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,10)=feederFAULT99(i+feederindex,10)+1;
          elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,10)=feederFAULT00(i+feederindex,10)+1;
          elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,10)=feederFAULT01(i+feederindex,10)+1;
          elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,10)=feederFAULT02(i+feederindex,10)+1;
          elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,10)=feederFAULT03(i+feederindex,10)+1;
          elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352
```

```
feederFAULT04(i+feederindex,10)=feederFAULT04(i+feederindex,10)+1;
            end
            feederFAULT(i+feederindex,10)=feederFAULT(i+feederindex,10)+1;
        end
        if reg_0(j,13)==54

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,11)=feederFAULT98(i+feederindex,11)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,11)=feederFAULT99(i+feederindex,11)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,11)=feederFAULT00(i+feederindex,11)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,11)=feederFAULT01(i+feederindex,11)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,11)=feederFAULT02(i+feederindex,11)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,11)=feederFAULT03(i+feederindex,11)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,11)=feederFAULT04(i+feederindex,11)+1;
            end
            feederFAULT(i+feederindex,11)=feederFAULT(i+feederindex,11)+1;
        end
        if reg_0(j,13)==56

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,12)=feederFAULT98(i+feederindex,12)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,12)=feederFAULT99(i+feederindex,12)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,12)=feederFAULT00(i+feederindex,12)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,12)=feederFAULT01(i+feederindex,12)+1;
```

```
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,12)=feederFAULT02(i+feederindex,12)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,12)=feederFAULT03(i+feederindex,12)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,12)=feederFAULT04(i+feederindex,12)+1;
            end
            feederFAULT(i+feederindex,12)=feederFAULT(i+feederindex,12)+1;
        end
        if reg_0(j,13)==58

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,13)=feederFAULT98(i+feederindex,13)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,13)=feederFAULT99(i+feederindex,13)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,13)=feederFAULT00(i+feederindex,13)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,13)=feederFAULT01(i+feederindex,13)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,13)=feederFAULT02(i+feederindex,13)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,13)=feederFAULT03(i+feederindex,13)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,13)=feederFAULT04(i+feederindex,13)+1;
            end
            feederFAULT(i+feederindex,13)=feederFAULT(i+feederindex,13)+1;
        end
        if reg_0(j,13)==59

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,14)=feederFAULT98(i+feederindex,14)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525
```

```
feederFAULT99(i+feederindex,14)=feederFAULT99(i+feederindex,14)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,14)=feederFAULT00(i+feederindex,14)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,14)=feederFAULT01(i+feederindex,14)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,14)=feederFAULT02(i+feederindex,14)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,14)=feederFAULT03(i+feederindex,14)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,14)=feederFAULT04(i+feederindex,14)+1;
            end
            feederFAULT(i+feederindex,14)=feederFAULT(i+feederindex,14)+1;
        end
        if reg_0(j,13)==63

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,15)=feederFAULT98(i+feederindex,15)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,15)=feederFAULT99(i+feederindex,15)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,15)=feederFAULT00(i+feederindex,15)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,15)=feederFAULT01(i+feederindex,15)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,15)=feederFAULT02(i+feederindex,15)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,15)=feederFAULT03(i+feederindex,15)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,15)=feederFAULT04(i+feederindex,15)+1;
            end
            feederFAULT(i+feederindex,15)=feederFAULT(i+feederindex,15)+1;
```

```
            end
            if reg_0(j,13)==65

               if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,16)=feederFAULT98(i+feederindex,16)+1;
               elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,16)=feederFAULT99(i+feederindex,16)+1;
               elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,16)=feederFAULT00(i+feederindex,16)+1;
               elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,16)=feederFAULT01(i+feederindex,16)+1;
               elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,16)=feederFAULT02(i+feederindex,16)+1;
               elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,16)=feederFAULT03(i+feederindex,16)+1;
               elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,16)=feederFAULT04(i+feederindex,16)+1;
               end
               feederFAULT(i+feederindex,16)=feederFAULT(i+feederindex,16)+1;
            end
            if reg_0(j,13)==66

               if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,17)=feederFAULT98(i+feederindex,17)+1;
               elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,17)=feederFAULT99(i+feederindex,17)+1;
               elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,17)=feederFAULT00(i+feederindex,17)+1;
               elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,17)=feederFAULT01(i+feederindex,17)+1;
               elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,17)=feederFAULT02(i+feederindex,17)+1;
               elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986
```

```
feederFAULT03(i+feederindex,17)=feederFAULT03(i+feederindex,17)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,17)=feederFAULT04(i+feederindex,17)+1;
            end
            feederFAULT(i+feederindex,17)=feederFAULT(i+feederindex,17)+1;
        end
        if reg_0(j,13)==67

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,18)=feederFAULT98(i+feederindex,18)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,18)=feederFAULT99(i+feederindex,18)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,18)=feederFAULT00(i+feederindex,18)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,18)=feederFAULT01(i+feederindex,18)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,18)=feederFAULT02(i+feederindex,18)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,18)=feederFAULT03(i+feederindex,18)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,18)=feederFAULT04(i+feederindex,18)+1;
            end
            feederFAULT(i+feederindex,18)=feederFAULT(i+feederindex,18)+1;
        end
        if reg_0(j,13)==69

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,19)=feederFAULT98(i+feederindex,19)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,19)=feederFAULT99(i+feederindex,19)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,19)=feederFAULT00(i+feederindex,19)+1;
```

```
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,19)=feederFAULT01(i+feederindex,19)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,19)=feederFAULT02(i+feederindex,19)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,19)=feederFAULT03(i+feederindex,19)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,19)=feederFAULT04(i+feederindex,19)+1;
            end
            feederFAULT(i+feederindex,19)=feederFAULT(i+feederindex,19)+1;
        end
        if reg_0(j,13)==83

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,20)=feederFAULT98(i+feederindex,20)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,20)=feederFAULT99(i+feederindex,20)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,20)=feederFAULT00(i+feederindex,20)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,20)=feederFAULT01(i+feederindex,20)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,20)=feederFAULT02(i+feederindex,20)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,20)=feederFAULT03(i+feederindex,20)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,20)=feederFAULT04(i+feederindex,20)+1;
            end
            feederFAULT(i+feederindex,20)=feederFAULT(i+feederindex,20)+1;
        end
        if reg_0(j,13)==85

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
```

```
feederFAULT98(i+feederindex,21)=feederFAULT98(i+feederindex,21)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,21)=feederFAULT99(i+feederindex,21)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,21)=feederFAULT00(i+feederindex,21)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,21)=feederFAULT01(i+feederindex,21)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,21)=feederFAULT02(i+feederindex,21)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,21)=feederFAULT03(i+feederindex,21)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,21)=feederFAULT04(i+feederindex,21)+1;
            end
            feederFAULT(i+feederindex,21)=feederFAULT(i+feederindex,21)+1;
        end
        if reg_0(j,13)==86

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,22)=feederFAULT98(i+feederindex,22)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,22)=feederFAULT99(i+feederindex,22)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,22)=feederFAULT00(i+feederindex,22)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,22)=feederFAULT01(i+feederindex,22)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,22)=feederFAULT02(i+feederindex,22)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,22)=feederFAULT03(i+feederindex,22)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352
```

```
feederFAULT04(i+feederindex,22)=feederFAULT04(i+feederindex,22)+1;
            end
            feederFAULT(i+feederindex,22)=feederFAULT(i+feederindex,22)+1;
        end
        if reg_0(j,13)==87

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,23)=feederFAULT98(i+feederindex,23)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,23)=feederFAULT99(i+feederindex,23)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,23)=feederFAULT00(i+feederindex,23)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,23)=feederFAULT01(i+feederindex,23)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,23)=feederFAULT02(i+feederindex,23)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,23)=feederFAULT03(i+feederindex,23)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,23)=feederFAULT04(i+feederindex,23)+1;
            end
            feederFAULT(i+feederindex,23)=feederFAULT(i+feederindex,23)+1;
        end
        if reg_0(j,13)==88

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,24)=feederFAULT98(i+feederindex,24)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,24)=feederFAULT99(i+feederindex,24)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,24)=feederFAULT00(i+feederindex,24)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,24)=feederFAULT01(i+feederindex,24)+1;
```

```
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,24)=feederFAULT02(i+feederindex,24)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,24)=feederFAULT03(i+feederindex,24)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,24)=feederFAULT04(i+feederindex,24)+1;
            end
            feederFAULT(i+feederindex,24)=feederFAULT(i+feederindex,24)+1;
        end
        if reg_0(j,13)==89

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,25)=feederFAULT98(i+feederindex,25)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,25)=feederFAULT99(i+feederindex,25)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,25)=feederFAULT00(i+feederindex,25)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,25)=feederFAULT01(i+feederindex,25)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,25)=feederFAULT02(i+feederindex,25)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,25)=feederFAULT03(i+feederindex,25)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,25)=feederFAULT04(i+feederindex,25)+1;
            end
            feederFAULT(i+feederindex,25)=feederFAULT(i+feederindex,25)+1;
        end
        if reg_0(j,13)==90

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,26)=feederFAULT98(i+feederindex,26)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525
```

```
feederFAULT99(i+feederindex,26)=feederFAULT99(i+feederindex,26)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,26)=feederFAULT00(i+feederindex,26)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,26)=feederFAULT01(i+feederindex,26)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,26)=feederFAULT02(i+feederindex,26)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,26)=feederFAULT03(i+feederindex,26)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,26)=feederFAULT04(i+feederindex,26)+1;
            end
            feederFAULT(i+feederindex,26)=feederFAULT(i+feederindex,26)+1;
        end
        if reg_0(j,13)==91

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,27)=feederFAULT98(i+feederindex,27)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,27)=feederFAULT99(i+feederindex,27)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,27)=feederFAULT00(i+feederindex,27)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,27)=feederFAULT01(i+feederindex,27)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,27)=feederFAULT02(i+feederindex,27)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,27)=feederFAULT03(i+feederindex,27)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,27)=feederFAULT04(i+feederindex,27)+1;
            end
            feederFAULT(i+feederindex,27)=feederFAULT(i+feederindex,27)+1;
```

```
            end
            if reg_0(j,13)==92

               if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,28)=feederFAULT98(i+feederindex,28)+1;
               elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,28)=feederFAULT99(i+feederindex,28)+1;
               elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,28)=feederFAULT00(i+feederindex,28)+1;
               elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,28)=feederFAULT01(i+feederindex,28)+1;
               elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,28)=feederFAULT02(i+feederindex,28)+1;
               elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,28)=feederFAULT03(i+feederindex,28)+1;
               elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,28)=feederFAULT04(i+feederindex,28)+1;
               end
               feederFAULT(i+feederindex,28)=feederFAULT(i+feederindex,28)+1;
            end
            if reg_0(j,13)==93

               if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,29)=feederFAULT98(i+feederindex,29)+1;
               elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,29)=feederFAULT99(i+feederindex,29)+1;
               elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,29)=feederFAULT00(i+feederindex,29)+1;
               elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,29)=feederFAULT01(i+feederindex,29)+1;
               elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,29)=feederFAULT02(i+feederindex,29)+1;
               elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986
```

```
feederFAULT03(i+feederindex,29)=feederFAULT03(i+feederindex,29)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,29)=feederFAULT04(i+feederindex,29)+1;
            end
            feederFAULT(i+feederindex,29)=feederFAULT(i+feederindex,29)+1;
        end
        if reg_0(j,13)==94

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,30)=feederFAULT98(i+feederindex,30)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,30)=feederFAULT99(i+feederindex,30)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,30)=feederFAULT00(i+feederindex,30)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,30)=feederFAULT01(i+feederindex,30)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,30)=feederFAULT02(i+feederindex,30)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,30)=feederFAULT03(i+feederindex,30)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,30)=feederFAULT04(i+feederindex,30)+1;
            end
            feederFAULT(i+feederindex,30)=feederFAULT(i+feederindex,30)+1;
        end
        if reg_0(j,13)==95

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,31)=feederFAULT98(i+feederindex,31)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,31)=feederFAULT99(i+feederindex,31)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,31)=feederFAULT00(i+feederindex,31)+1;
```

```
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,31)=feederFAULT01(i+feederindex,31)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,31)=feederFAULT02(i+feederindex,31)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,31)=feederFAULT03(i+feederindex,31)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,31)=feederFAULT04(i+feederindex,31)+1;
            end
            feederFAULT(i+feederindex,31)=feederFAULT(i+feederindex,31)+1;
        end
        if reg_0(j,13)==96

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,32)=feederFAULT98(i+feederindex,32)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,32)=feederFAULT99(i+feederindex,32)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,32)=feederFAULT00(i+feederindex,32)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,32)=feederFAULT01(i+feederindex,32)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,32)=feederFAULT02(i+feederindex,32)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,32)=feederFAULT03(i+feederindex,32)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,32)=feederFAULT04(i+feederindex,32)+1;
            end
            feederFAULT(i+feederindex,32)=feederFAULT(i+feederindex,32)+1;
        end
        if reg_0(j,13)==97

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
```

```
feederFAULT98(i+feederindex,33)=feederFAULT98(i+feederindex,33)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,33)=feederFAULT99(i+feederindex,33)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,33)=feederFAULT00(i+feederindex,33)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,33)=feederFAULT01(i+feederindex,33)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,33)=feederFAULT02(i+feederindex,33)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,33)=feederFAULT03(i+feederindex,33)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,33)=feederFAULT04(i+feederindex,33)+1;
            end
            feederFAULT(i+feederindex,33)=feederFAULT(i+feederindex,33)+1;
        end
        if reg_0(j,13)==98

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,34)=feederFAULT98(i+feederindex,34)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,34)=feederFAULT99(i+feederindex,34)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,34)=feederFAULT00(i+feederindex,34)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,34)=feederFAULT01(i+feederindex,34)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,34)=feederFAULT02(i+feederindex,34)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,34)=feederFAULT03(i+feederindex,34)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352
```

```
feederFAULT04(i+feederindex,34)=feederFAULT04(i+feederindex,34)+1;
            end
            feederFAULT(i+feederindex,34)=feederFAULT(i+feederindex,34)+1;
        end
        if reg_0(j,13)==99

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,35)=feederFAULT98(i+feederindex,35)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,35)=feederFAULT99(i+feederindex,35)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,35)=feederFAULT00(i+feederindex,35)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,35)=feederFAULT01(i+feederindex,35)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,35)=feederFAULT02(i+feederindex,35)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,35)=feederFAULT03(i+feederindex,35)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,35)=feederFAULT04(i+feederindex,35)+1;
            end
            feederFAULT(i+feederindex,35)=feederFAULT(i+feederindex,35)+1;
        end
        if reg_0(j,9)==1

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,39)=feederFAULT98(i+feederindex,39)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,39)=feederFAULT99(i+feederindex,39)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,39)=feederFAULT00(i+feederindex,39)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,39)=feederFAULT01(i+feederindex,39)+1;
```

```
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,39)=feederFAULT02(i+feederindex,39)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,39)=feederFAULT03(i+feederindex,39)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,39)=feederFAULT04(i+feederindex,39)+1;
            end
            feederFAULT(i+feederindex,39)=feederFAULT(i+feederindex,39)+1;
          end
          if reg_0(j,9)==2

            if reg_0(j,1)>=35796 && reg_0(j,1)<=38352

            end
            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160


feederFAULT98(i+feederindex,40)=feederFAULT98(i+feederindex,40)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525


feederFAULT99(i+feederindex,40)=feederFAULT99(i+feederindex,40)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891


feederFAULT00(i+feederindex,40)=feederFAULT00(i+feederindex,40)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256


feederFAULT01(i+feederindex,40)=feederFAULT01(i+feederindex,40)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621


feederFAULT02(i+feederindex,40)=feederFAULT02(i+feederindex,40)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986


feederFAULT03(i+feederindex,40)=feederFAULT03(i+feederindex,40)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352


feederFAULT04(i+feederindex,40)=feederFAULT04(i+feederindex,40)+1;
```

```
            end
            feederFAULT(i+feederindex,40)=feederFAULT(i+feederindex,40)+1;
        end
        if reg_0(j,9)==3

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,41)=feederFAULT98(i+feederindex,41)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,41)=feederFAULT99(i+feederindex,41)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,41)=feederFAULT00(i+feederindex,41)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,41)=feederFAULT01(i+feederindex,41)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,41)=feederFAULT02(i+feederindex,41)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,41)=feederFAULT03(i+feederindex,41)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,41)=feederFAULT04(i+feederindex,41)+1;
            end
            feederFAULT(i+feederindex,41)=feederFAULT(i+feederindex,41)+1;
        end
        if reg_0(j,14)==1

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,43)=feederFAULT98(i+feederindex,43)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,43)=feederFAULT99(i+feederindex,43)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,43)=feederFAULT00(i+feederindex,43)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,43)=feederFAULT01(i+feederindex,43)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621
```

```
feederFAULT02(i+feederindex,43)=feederFAULT02(i+feederindex,43)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,43)=feederFAULT03(i+feederindex,43)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,43)=feederFAULT04(i+feederindex,43)+1;
            end
            feederFAULT(i+feederindex,43)=feederFAULT(i+feederindex,43)+1;
          end
          if reg_0(j,14)==1 && reg_0(j,15)==1 %Riser fault count

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160


feederFAULT98(i+feederindex,44)=feederFAULT98(i+feederindex,44)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,44)=feederFAULT99(i+feederindex,44)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,44)=feederFAULT00(i+feederindex,44)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,44)=feederFAULT01(i+feederindex,44)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,44)=feederFAULT02(i+feederindex,44)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,44)=feederFAULT03(i+feederindex,44)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,44)=feederFAULT04(i+feederindex,44)+1;
            end
            feederFAULT(i+feederindex,44)=feederFAULT(i+feederindex,44)+1;
          end
        end

    end
```

```
if i==feedernames_size(1,1) %If added to fix the problem when the last feeder was read on feedernames, which had no next feeder to use as end index in the previous IF.

    for j=feedernames(feedernames_size(1,1),2):reg_0_size(1,1)


        % SAIDI----------use clients(row 12) and duration (row 7)
        if reg_0(j,7)>momentary %If prevents counting reclosings as part of SAIDI

            %Division by #clients will be moved inside this if
            %because: 1. Sum of ratios == a ratio of sums
            %2. THis way i can divide by the normal number of total
            %clients or those specified in the database, which
            %could be higher due to use of interties.



            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                    clients=reg_0(j,12);
                else
                    clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                    clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));
                if clients>0
                    saididailytemp(j,1)=reg_0(j,11);
                    saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                    saididailytemp(j,3)=clients;
                    saididailytemp(j,4)=saididailynumonly;
                end
                if (saididaily>=feederMEDtuki(i+feederindex,4))&&(r~=4)
                    saididaily=0;
                    saididailynumonly=0;
                    MEDcount98=MEDcount98+1;
                end
                if clients<=50
                    saididaily=0;
                    saididailynumonly=0;
                end
                if reg_0(j,13)==38 || reg_0(j,13)==39
```

```
                saididaily=0;
                saididailynumonly=0;

            end

feederoutputMEDDED(i+feederindex,8)=saididaily+feederoutputMEDDED(i+feederind
ex,8); %SAIDI

regdisoutputMEDDED(i+feederindex,8)=saididailynumonly+regdisoutputMEDDED(i+f
eederindex,8);
            if clients>0
                saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytempMEDDED(j,3)=clients;
                saididailytempMEDDED(j,4)=saididailynumonly;
            end


        end
        if reg_0(j,1)>=36161 && reg_0(j,1)<=36525
            if reg_0(j,12)>feederoutputMEDDED(i+feederindex,5)
                clients=reg_0(j,12);
            else
                clients=feederoutputMEDDED(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end

            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
            saididailynumonly=(reg_0(j,12)*reg_0(j,7));
            if clients>0
                saididailytemp(j,1)=reg_0(j,11);
                saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytemp(j,3)=clients;
                saididailytemp(j,5)=saididailynumonly;
            end
            if saididaily>=feederMEDtuki(i+feederindex,6)
                saididaily=0;
                saididailynumonly=0;
                MEDcount99=MEDcount99+1;
            end
            if clients<=50
                saididaily=0;
```

```
                saididailynumonly=0;
            end
            if reg_0(j,13)==38 || reg_0(j,13)==39
                saididaily=0;
                saididailynumonly=0;

            end

feederoutputMEDDED(i+feederindex,9)=saididaily+feederoutputMEDDED(i+feederind
ex,9); %SAIDI

regdisoutputMEDDED(i+feederindex,9)=saididailynumonly+regdisoutputMEDDED(i+f
eederindex,9);
            if clients>0
                saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytempMEDDED(j,3)=clients;
                saididailytempMEDDED(j,5)=saididailynumonly;
            end


        end
        if reg_0(j,1)>=36526 && reg_0(j,1)<=36891
            if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                clients=reg_0(j,12);
            else
                clients=feederoutputMEDDED(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end

            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
            saididailynumonly=(reg_0(j,12)*reg_0(j,7));
            if clients>0
                saididailytemp(j,1)=reg_0(j,11);
                saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytemp(j,3)=clients;
                saididailytemp(j,6)=saididailynumonly;
            end
            if saididaily>=feederMEDtuki(i+feederindex,7)
                saididaily=0;
                saididailynumonly=0;
                MEDcount00=MEDcount00+1;
```

```
                end
                if clients<=50
                   saididaily=0;
                   saididailynumonly=0;
                end
                if reg_0(j,13)==38 || reg_0(j,13)==39
                   saididaily=0;
                   saididailynumonly=0;

                end

feederoutputMEDDED(i+feederindex,10)=saididaily+feederoutputMEDDED(i+feederin
dex,10); %SAIDI

regdisoutputMEDDED(i+feederindex,10)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,10);
                if clients>0
                   saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytempMEDDED(j,3)=clients;
                   saididailytempMEDDED(j,6)=saididailynumonly;
                end


             end
             if reg_0(j,1)>=36892 && reg_0(j,1)<=37256
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                   clients=reg_0(j,12);
                else
                   clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                   clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));
                if clients>0
                   saididailytemp(j,1)=reg_0(j,11);
                   saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytemp(j,3)=clients;
                   saididailytemp(j,7)=saididailynumonly;
                end
                if saididaily>=feederMEDtuki(i+feederindex,8)
```

```
                saididaily=0;
                saididailynumonly=0;
                MEDcount01=MEDcount01+1;
            end
            if clients<=50
                saididaily=0;
                saididailynumonly=0;
            end
            if reg_0(j,13)==38 || reg_0(j,13)==39
                saididaily=0;
                saididailynumonly=0;

            end

feederoutputMEDDED(i+feederindex,11)=saididaily+feederoutputMEDDED(i+feederin
dex,11); %SAIDI

regdisoutputMEDDED(i+feederindex,11)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,11);
            if clients>0
                saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytempMEDDED(j,3)=clients;
                saididailytempMEDDED(j,7)=saididailynumonly;
            end



        end
        if reg_0(j,1)>=37257 && reg_0(j,1)<=37621
            if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                clients=reg_0(j,12);
            else
                clients=feederoutputMEDDED(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end

            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
            saididailynumonly=(reg_0(j,12)*reg_0(j,7));
            if clients>0
                saididailytemp(j,1)=reg_0(j,11);
                saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
```

```
                saididailytemp(j,3)=clients;
                saididailytemp(j,8)=saididailynumonly;
            end
            if saididaily>=feederMEDtuki(i+feederindex,9)
                saididaily=0;
                saididailynumonly=0;
                MEDcount02=MEDcount02+1;
            end
            if clients<=50
                saididaily=0;
                saididailynumonly=0;
            end
            if reg_0(j,13)==38 || reg_0(j,13)==39
                saididaily=0;
                saididailynumonly=0;

            end

feederoutputMEDDED(i+feederindex,12)=saididaily+feederoutputMEDDED(i+feederin
dex,12); %SAIDI

regdisoutputMEDDED(i+feederindex,12)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,12);
            if clients>0
                saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytempMEDDED(j,3)=clients;
                saididailytempMEDDED(j,8)=saididailynumonly;
            end



        end
        if reg_0(j,1)>=37622 && reg_0(j,1)<=37986
            if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                clients=reg_0(j,12);
            else
                clients=feederoutputMEDDED(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end

            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
```

```
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));
                if clients>0
                   saididailytemp(j,1)=reg_0(j,11);
                   saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytemp(j,3)=clients;
                   saididailytemp(j,9)=saididailynumonly;
                end
                if saididaily>=feederMEDtuki(i+feederindex,10)
                   saididaily=0;
                   saididailynumonly=0;
                   MEDcount03=MEDcount03+1;
                end
                if clients<=50
                   saididaily=0;
                   saididailynumonly=0;
                end
                if reg_0(j,13)==38 || reg_0(j,13)==39
                   saididaily=0;
                   saididailynumonly=0;

                end

feederoutputMEDDED(i+feederindex,13)=saididaily+feederoutputMEDDED(i+feederin
dex,13); %SAIDI

regdisoutputMEDDED(i+feederindex,13)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,13);
                if clients>0
                   saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytempMEDDED(j,3)=clients;
                   saididailytempMEDDED(j,9)=saididailynumonly;
                end



             end
             if reg_0(j,1)>=37987 && reg_0(j,1)<=38352
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                   clients=reg_0(j,12);
                else
                   clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
```

```
                clients=99;
            end

            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
            saididailynumonly=(reg_0(j,12)*reg_0(j,7));
            if clients>0
                saididailytemp(j,1)=reg_0(j,11);
                saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytemp(j,3)=clients;
                saididailytemp(j,10)=saididailynumonly;
            end
            if saididaily>=feederMEDtuki(i+feederindex,9)
                saididaily=0;
                saididailynumonly=0;
                MEDcount04=MEDcount04+1;
            end
            if clients<=50
                saididaily=0;
                saididailynumonly=0;
            end
            if reg_0(j,13)==38 || reg_0(j,13)==39
                saididaily=0;
                saididailynumonly=0;

            end

feederoutputMEDDED(i+feederindex,14)=saididaily+feederoutputMEDDED(i+feederin
dex,14); %SAIDI

regdisoutputMEDDED(i+feederindex,14)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,14);
            if clients>0
                saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytempMEDDED(j,3)=clients;
                saididailytempMEDDED(j,10)=saididailynumonly;
            end



        end

    end
    %FAULT EXTRACTOR V2
```

```
if reg_0(j,13)==13

   if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
      feederFAULT98(i+feederindex,4)=feederFAULT98(i+feederindex,4)+1;
   elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525
      feederFAULT99(i+feederindex,4)=feederFAULT99(i+feederindex,4)+1;
   elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891
      feederFAULT00(i+feederindex,4)=feederFAULT00(i+feederindex,4)+1;
   elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256
      feederFAULT01(i+feederindex,4)=feederFAULT01(i+feederindex,4)+1;
   elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621
      feederFAULT02(i+feederindex,4)=feederFAULT02(i+feederindex,4)+1;
   elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986
      feederFAULT03(i+feederindex,4)=feederFAULT03(i+feederindex,4)+1;
   elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352
      feederFAULT04(i+feederindex,4)=feederFAULT04(i+feederindex,4)+1;
   end

   feederFAULT(i+feederindex,4)=feederFAULT(i+feederindex,4)+1;
end
if reg_0(j,13)==38

   if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

      feederFAULT98(i+feederindex,5)=feederFAULT98(i+feederindex,5)+1;
   elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

      feederFAULT99(i+feederindex,5)=feederFAULT99(i+feederindex,5)+1;
   elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

      feederFAULT00(i+feederindex,5)=feederFAULT00(i+feederindex,5)+1;
   elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

      feederFAULT01(i+feederindex,5)=feederFAULT01(i+feederindex,5)+1;
   elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

      feederFAULT02(i+feederindex,5)=feederFAULT02(i+feederindex,5)+1;
   elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

      feederFAULT03(i+feederindex,5)=feederFAULT03(i+feederindex,5)+1;
   elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

      feederFAULT04(i+feederindex,5)=feederFAULT04(i+feederindex,5)+1;
   end
```

```
   feederFAULT(i+feederindex,5)=feederFAULT(i+feederindex,5)+1;
end
if reg_0(j,13)==39

   if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
      feederFAULT98(i+feederindex,6)=feederFAULT98(i+feederindex,6)+1;
   elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525
      feederFAULT99(i+feederindex,6)=feederFAULT99(i+feederindex,6)+1;
   elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891
      feederFAULT00(i+feederindex,6)=feederFAULT00(i+feederindex,6)+1;
   elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256
      feederFAULT01(i+feederindex,6)=feederFAULT01(i+feederindex,6)+1;
   elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621
      feederFAULT02(i+feederindex,6)=feederFAULT02(i+feederindex,6)+1;
   elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986
      feederFAULT03(i+feederindex,6)=feederFAULT03(i+feederindex,6)+1;
   elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352
      feederFAULT04(i+feederindex,6)=feederFAULT04(i+feederindex,6)+1;
   end
   feederFAULT(i+feederindex,6)=feederFAULT(i+feederindex,6)+1;
end
if reg_0(j,13)==48


   if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

      feederFAULT98(i+feederindex,7)=feederFAULT98(i+feederindex,7)+1;
   elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

      feederFAULT99(i+feederindex,7)=feederFAULT99(i+feederindex,7)+1;
   elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

      feederFAULT00(i+feederindex,7)=feederFAULT00(i+feederindex,7)+1;
   elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

      feederFAULT01(i+feederindex,7)=feederFAULT01(i+feederindex,7)+1;
   elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

      feederFAULT02(i+feederindex,7)=feederFAULT02(i+feederindex,7)+1;
   elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

      feederFAULT03(i+feederindex,7)=feederFAULT03(i+feederindex,7)+1;
   elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

      feederFAULT04(i+feederindex,7)=feederFAULT04(i+feederindex,7)+1;
```

```
    end
    feederFAULT(i+feederindex,7)=feederFAULT(i+feederindex,7)+1;
end
if reg_0(j,13)==51

    if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
        feederFAULT98(i+feederindex,8)=feederFAULT98(i+feederindex,8)+1;
    elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525
        feederFAULT99(i+feederindex,8)=feederFAULT99(i+feederindex,8)+1;
    elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891
        feederFAULT00(i+feederindex,8)=feederFAULT00(i+feederindex,8)+1;
    elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256
        feederFAULT01(i+feederindex,8)=feederFAULT01(i+feederindex,8)+1;
    elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621
        feederFAULT02(i+feederindex,8)=feederFAULT02(i+feederindex,8)+1;
    elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986
        feederFAULT03(i+feederindex,8)=feederFAULT03(i+feederindex,8)+1;
    elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352
        feederFAULT04(i+feederindex,8)=feederFAULT04(i+feederindex,8)+1;
    end
    feederFAULT(i+feederindex,8)=feederFAULT(i+feederindex,8)+1;
end
if reg_0(j,13)==52

    if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
        feederFAULT98(i+feederindex,9)=feederFAULT98(i+feederindex,9)+1;
    elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525
        feederFAULT99(i+feederindex,9)=feederFAULT99(i+feederindex,9)+1;
    elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891
        feederFAULT00(i+feederindex,9)=feederFAULT00(i+feederindex,9)+1;
    elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256
        feederFAULT01(i+feederindex,9)=feederFAULT01(i+feederindex,9)+1;
    elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621
        feederFAULT02(i+feederindex,9)=feederFAULT02(i+feederindex,9)+1;
    elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986
        feederFAULT03(i+feederindex,9)=feederFAULT03(i+feederindex,9)+1;
    elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352
        feederFAULT04(i+feederindex,9)=feederFAULT04(i+feederindex,9)+1;
    end
    feederFAULT(i+feederindex,9)=feederFAULT(i+feederindex,9)+1;
end
if reg_0(j,13)==53

    if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
```

```
feederFAULT98(i+feederindex,10)=feederFAULT98(i+feederindex,10)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,10)=feederFAULT99(i+feederindex,10)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,10)=feederFAULT00(i+feederindex,10)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,10)=feederFAULT01(i+feederindex,10)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,10)=feederFAULT02(i+feederindex,10)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,10)=feederFAULT03(i+feederindex,10)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,10)=feederFAULT04(i+feederindex,10)+1;
            end
            feederFAULT(i+feederindex,10)=feederFAULT(i+feederindex,10)+1;
        end
        if reg_0(j,13)==54

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,11)=feederFAULT98(i+feederindex,11)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,11)=feederFAULT99(i+feederindex,11)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,11)=feederFAULT00(i+feederindex,11)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,11)=feederFAULT01(i+feederindex,11)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,11)=feederFAULT02(i+feederindex,11)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,11)=feederFAULT03(i+feederindex,11)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352
```

```
feederFAULT04(i+feederindex,11)=feederFAULT04(i+feederindex,11)+1;
            end
            feederFAULT(i+feederindex,11)=feederFAULT(i+feederindex,11)+1;
        end
        if reg_0(j,13)==56

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,12)=feederFAULT98(i+feederindex,12)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,12)=feederFAULT99(i+feederindex,12)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,12)=feederFAULT00(i+feederindex,12)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,12)=feederFAULT01(i+feederindex,12)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,12)=feederFAULT02(i+feederindex,12)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,12)=feederFAULT03(i+feederindex,12)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,12)=feederFAULT04(i+feederindex,12)+1;
            end
            feederFAULT(i+feederindex,12)=feederFAULT(i+feederindex,12)+1;
        end
        if reg_0(j,13)==58

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,13)=feederFAULT98(i+feederindex,13)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,13)=feederFAULT99(i+feederindex,13)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,13)=feederFAULT00(i+feederindex,13)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,13)=feederFAULT01(i+feederindex,13)+1;
```

```
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,13)=feederFAULT02(i+feederindex,13)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,13)=feederFAULT03(i+feederindex,13)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,13)=feederFAULT04(i+feederindex,13)+1;
            end
            feederFAULT(i+feederindex,13)=feederFAULT(i+feederindex,13)+1;
        end
        if reg_0(j,13)==59

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,14)=feederFAULT98(i+feederindex,14)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,14)=feederFAULT99(i+feederindex,14)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,14)=feederFAULT00(i+feederindex,14)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,14)=feederFAULT01(i+feederindex,14)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,14)=feederFAULT02(i+feederindex,14)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,14)=feederFAULT03(i+feederindex,14)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,14)=feederFAULT04(i+feederindex,14)+1;
            end
            feederFAULT(i+feederindex,14)=feederFAULT(i+feederindex,14)+1;
        end
        if reg_0(j,13)==63

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,15)=feederFAULT98(i+feederindex,15)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525
```

```
feederFAULT99(i+feederindex,15)=feederFAULT99(i+feederindex,15)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,15)=feederFAULT00(i+feederindex,15)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,15)=feederFAULT01(i+feederindex,15)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,15)=feederFAULT02(i+feederindex,15)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,15)=feederFAULT03(i+feederindex,15)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,15)=feederFAULT04(i+feederindex,15)+1;
            end
            feederFAULT(i+feederindex,15)=feederFAULT(i+feederindex,15)+1;
        end
        if reg_0(j,13)==65

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,16)=feederFAULT98(i+feederindex,16)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,16)=feederFAULT99(i+feederindex,16)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,16)=feederFAULT00(i+feederindex,16)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,16)=feederFAULT01(i+feederindex,16)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,16)=feederFAULT02(i+feederindex,16)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,16)=feederFAULT03(i+feederindex,16)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,16)=feederFAULT04(i+feederindex,16)+1;
            end
            feederFAULT(i+feederindex,16)=feederFAULT(i+feederindex,16)+1;
```

```
        end
        if reg_0(j,13)==66

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,17)=feederFAULT98(i+feederindex,17)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,17)=feederFAULT99(i+feederindex,17)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,17)=feederFAULT00(i+feederindex,17)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,17)=feederFAULT01(i+feederindex,17)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,17)=feederFAULT02(i+feederindex,17)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,17)=feederFAULT03(i+feederindex,17)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,17)=feederFAULT04(i+feederindex,17)+1;
            end
            feederFAULT(i+feederindex,17)=feederFAULT(i+feederindex,17)+1;
        end
        if reg_0(j,13)==67

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,18)=feederFAULT98(i+feederindex,18)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,18)=feederFAULT99(i+feederindex,18)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,18)=feederFAULT00(i+feederindex,18)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,18)=feederFAULT01(i+feederindex,18)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,18)=feederFAULT02(i+feederindex,18)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986
```

```
feederFAULT03(i+feederindex,18)=feederFAULT03(i+feederindex,18)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,18)=feederFAULT04(i+feederindex,18)+1;
            end
            feederFAULT(i+feederindex,18)=feederFAULT(i+feederindex,18)+1;
        end
        if reg_0(j,13)==69

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,19)=feederFAULT98(i+feederindex,19)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,19)=feederFAULT99(i+feederindex,19)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,19)=feederFAULT00(i+feederindex,19)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,19)=feederFAULT01(i+feederindex,19)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,19)=feederFAULT02(i+feederindex,19)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,19)=feederFAULT03(i+feederindex,19)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,19)=feederFAULT04(i+feederindex,19)+1;
            end
            feederFAULT(i+feederindex,19)=feederFAULT(i+feederindex,19)+1;
        end
        if reg_0(j,13)==83

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,20)=feederFAULT98(i+feederindex,20)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,20)=feederFAULT99(i+feederindex,20)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,20)=feederFAULT00(i+feederindex,20)+1;
```

```
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,20)=feederFAULT01(i+feederindex,20)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,20)=feederFAULT02(i+feederindex,20)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,20)=feederFAULT03(i+feederindex,20)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,20)=feederFAULT04(i+feederindex,20)+1;
            end
            feederFAULT(i+feederindex,20)=feederFAULT(i+feederindex,20)+1;
        end
        if reg_0(j,13)==85

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,21)=feederFAULT98(i+feederindex,21)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,21)=feederFAULT99(i+feederindex,21)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,21)=feederFAULT00(i+feederindex,21)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,21)=feederFAULT01(i+feederindex,21)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,21)=feederFAULT02(i+feederindex,21)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,21)=feederFAULT03(i+feederindex,21)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,21)=feederFAULT04(i+feederindex,21)+1;
            end
            feederFAULT(i+feederindex,21)=feederFAULT(i+feederindex,21)+1;
        end
        if reg_0(j,13)==86

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
```

```
feederFAULT98(i+feederindex,22)=feederFAULT98(i+feederindex,22)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,22)=feederFAULT99(i+feederindex,22)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,22)=feederFAULT00(i+feederindex,22)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,22)=feederFAULT01(i+feederindex,22)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,22)=feederFAULT02(i+feederindex,22)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,22)=feederFAULT03(i+feederindex,22)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,22)=feederFAULT04(i+feederindex,22)+1;
            end
            feederFAULT(i+feederindex,22)=feederFAULT(i+feederindex,22)+1;
        end
        if reg_0(j,13)==87

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,23)=feederFAULT98(i+feederindex,23)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,23)=feederFAULT99(i+feederindex,23)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,23)=feederFAULT00(i+feederindex,23)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,23)=feederFAULT01(i+feederindex,23)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,23)=feederFAULT02(i+feederindex,23)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,23)=feederFAULT03(i+feederindex,23)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352
```

```
feederFAULT04(i+feederindex,23)=feederFAULT04(i+feederindex,23)+1;
            end
            feederFAULT(i+feederindex,23)=feederFAULT(i+feederindex,23)+1;
        end
        if reg_0(j,13)==88

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,24)=feederFAULT98(i+feederindex,24)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,24)=feederFAULT99(i+feederindex,24)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,24)=feederFAULT00(i+feederindex,24)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,24)=feederFAULT01(i+feederindex,24)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,24)=feederFAULT02(i+feederindex,24)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,24)=feederFAULT03(i+feederindex,24)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,24)=feederFAULT04(i+feederindex,24)+1;
            end
            feederFAULT(i+feederindex,24)=feederFAULT(i+feederindex,24)+1;
        end
        if reg_0(j,13)==89

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,25)=feederFAULT98(i+feederindex,25)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,25)=feederFAULT99(i+feederindex,25)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,25)=feederFAULT00(i+feederindex,25)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,25)=feederFAULT01(i+feederindex,25)+1;
```

```
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,25)=feederFAULT02(i+feederindex,25)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,25)=feederFAULT03(i+feederindex,25)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,25)=feederFAULT04(i+feederindex,25)+1;
            end
            feederFAULT(i+feederindex,25)=feederFAULT(i+feederindex,25)+1;
        end
        if reg_0(j,13)==90

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,26)=feederFAULT98(i+feederindex,26)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,26)=feederFAULT99(i+feederindex,26)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,26)=feederFAULT00(i+feederindex,26)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,26)=feederFAULT01(i+feederindex,26)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,26)=feederFAULT02(i+feederindex,26)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,26)=feederFAULT03(i+feederindex,26)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,26)=feederFAULT04(i+feederindex,26)+1;
            end
            feederFAULT(i+feederindex,26)=feederFAULT(i+feederindex,26)+1;
        end
        if reg_0(j,13)==91

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,27)=feederFAULT98(i+feederindex,27)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525
```

```
feederFAULT99(i+feederindex,27)=feederFAULT99(i+feederindex,27)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,27)=feederFAULT00(i+feederindex,27)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,27)=feederFAULT01(i+feederindex,27)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,27)=feederFAULT02(i+feederindex,27)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,27)=feederFAULT03(i+feederindex,27)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,27)=feederFAULT04(i+feederindex,27)+1;
            end
            feederFAULT(i+feederindex,27)=feederFAULT(i+feederindex,27)+1;
        end
        if reg_0(j,13)==92

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,28)=feederFAULT98(i+feederindex,28)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,28)=feederFAULT99(i+feederindex,28)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,28)=feederFAULT00(i+feederindex,28)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,28)=feederFAULT01(i+feederindex,28)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,28)=feederFAULT02(i+feederindex,28)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,28)=feederFAULT03(i+feederindex,28)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,28)=feederFAULT04(i+feederindex,28)+1;
            end
            feederFAULT(i+feederindex,28)=feederFAULT(i+feederindex,28)+1;
```

```
            end
            if reg_0(j,13)==93

               if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,29)=feederFAULT98(i+feederindex,29)+1;
               elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,29)=feederFAULT99(i+feederindex,29)+1;
               elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,29)=feederFAULT00(i+feederindex,29)+1;
               elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,29)=feederFAULT01(i+feederindex,29)+1;
               elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,29)=feederFAULT02(i+feederindex,29)+1;
               elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,29)=feederFAULT03(i+feederindex,29)+1;
               elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,29)=feederFAULT04(i+feederindex,29)+1;
               end
               feederFAULT(i+feederindex,29)=feederFAULT(i+feederindex,29)+1;
            end
            if reg_0(j,13)==94

               if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,30)=feederFAULT98(i+feederindex,30)+1;
               elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,30)=feederFAULT99(i+feederindex,30)+1;
               elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,30)=feederFAULT00(i+feederindex,30)+1;
               elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,30)=feederFAULT01(i+feederindex,30)+1;
               elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,30)=feederFAULT02(i+feederindex,30)+1;
               elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986
```

```
feederFAULT03(i+feederindex,30)=feederFAULT03(i+feederindex,30)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,30)=feederFAULT04(i+feederindex,30)+1;
            end
            feederFAULT(i+feederindex,30)=feederFAULT(i+feederindex,30)+1;
        end
        if reg_0(j,13)==95

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,31)=feederFAULT98(i+feederindex,31)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,31)=feederFAULT99(i+feederindex,31)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,31)=feederFAULT00(i+feederindex,31)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,31)=feederFAULT01(i+feederindex,31)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,31)=feederFAULT02(i+feederindex,31)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,31)=feederFAULT03(i+feederindex,31)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,31)=feederFAULT04(i+feederindex,31)+1;
            end
            feederFAULT(i+feederindex,31)=feederFAULT(i+feederindex,31)+1;
        end
        if reg_0(j,13)==96

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,32)=feederFAULT98(i+feederindex,32)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,32)=feederFAULT99(i+feederindex,32)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,32)=feederFAULT00(i+feederindex,32)+1;
```

```
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,32)=feederFAULT01(i+feederindex,32)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,32)=feederFAULT02(i+feederindex,32)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,32)=feederFAULT03(i+feederindex,32)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,32)=feederFAULT04(i+feederindex,32)+1;
            end
            feederFAULT(i+feederindex,32)=feederFAULT(i+feederindex,32)+1;
        end
        if reg_0(j,13)==97

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,33)=feederFAULT98(i+feederindex,33)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,33)=feederFAULT99(i+feederindex,33)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,33)=feederFAULT00(i+feederindex,33)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,33)=feederFAULT01(i+feederindex,33)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,33)=feederFAULT02(i+feederindex,33)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,33)=feederFAULT03(i+feederindex,33)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,33)=feederFAULT04(i+feederindex,33)+1;
            end
            feederFAULT(i+feederindex,33)=feederFAULT(i+feederindex,33)+1;
        end
        if reg_0(j,13)==98

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
```

```
feederFAULT98(i+feederindex,34)=feederFAULT98(i+feederindex,34)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,34)=feederFAULT99(i+feederindex,34)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,34)=feederFAULT00(i+feederindex,34)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,34)=feederFAULT01(i+feederindex,34)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,34)=feederFAULT02(i+feederindex,34)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,34)=feederFAULT03(i+feederindex,34)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,34)=feederFAULT04(i+feederindex,34)+1;
            end
            feederFAULT(i+feederindex,34)=feederFAULT(i+feederindex,34)+1;
        end
        if reg_0(j,13)==99


            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,35)=feederFAULT98(i+feederindex,35)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,35)=feederFAULT99(i+feederindex,35)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,35)=feederFAULT00(i+feederindex,35)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,35)=feederFAULT01(i+feederindex,35)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,35)=feederFAULT02(i+feederindex,35)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,35)=feederFAULT03(i+feederindex,35)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352
```

```
feederFAULT04(i+feederindex,35)=feederFAULT04(i+feederindex,35)+1;
            end
            feederFAULT(i+feederindex,35)=feederFAULT(i+feederindex,35)+1;
        end
        if reg_0(j,9)==1

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,39)=feederFAULT98(i+feederindex,39)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,39)=feederFAULT99(i+feederindex,39)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,39)=feederFAULT00(i+feederindex,39)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,39)=feederFAULT01(i+feederindex,39)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,39)=feederFAULT02(i+feederindex,39)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,39)=feederFAULT03(i+feederindex,39)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,39)=feederFAULT04(i+feederindex,39)+1;
            end
            feederFAULT(i+feederindex,39)=feederFAULT(i+feederindex,39)+1;
        end
        if reg_0(j,9)==2

            if reg_0(j,1)>=35796 && reg_0(j,1)<=38352

            end


            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160


feederFAULT98(i+feederindex,40)=feederFAULT98(i+feederindex,40)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525
```

```
feederFAULT99(i+feederindex,40)=feederFAULT99(i+feederindex,40)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,40)=feederFAULT00(i+feederindex,40)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,40)=feederFAULT01(i+feederindex,40)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,40)=feederFAULT02(i+feederindex,40)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,40)=feederFAULT03(i+feederindex,40)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,40)=feederFAULT04(i+feederindex,40)+1;
            end
            feederFAULT(i+feederindex,40)=feederFAULT(i+feederindex,40)+1;
        end
        if reg_0(j,9)==3

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160

feederFAULT98(i+feederindex,41)=feederFAULT98(i+feederindex,41)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,41)=feederFAULT99(i+feederindex,41)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,41)=feederFAULT00(i+feederindex,41)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,41)=feederFAULT01(i+feederindex,41)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,41)=feederFAULT02(i+feederindex,41)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,41)=feederFAULT03(i+feederindex,41)+1;
```

```
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,41)=feederFAULT04(i+feederindex,41)+1;
            end
            feederFAULT(i+feederindex,41)=feederFAULT(i+feederindex,41)+1;
        end
        if reg_0(j,14)==1

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160


feederFAULT98(i+feederindex,43)=feederFAULT98(i+feederindex,43)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,43)=feederFAULT99(i+feederindex,43)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,43)=feederFAULT00(i+feederindex,43)+1;
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,43)=feederFAULT01(i+feederindex,43)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,43)=feederFAULT02(i+feederindex,43)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,43)=feederFAULT03(i+feederindex,43)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,43)=feederFAULT04(i+feederindex,43)+1;
            end
            feederFAULT(i+feederindex,43)=feederFAULT(i+feederindex,43)+1;
        end
        if reg_0(j,14)==1 && reg_0(j,15)==1 %Riser fault count

            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160


feederFAULT98(i+feederindex,44)=feederFAULT98(i+feederindex,44)+1;
            elseif reg_0(j,1)>=36161 && reg_0(j,1)<=36525

feederFAULT99(i+feederindex,44)=feederFAULT99(i+feederindex,44)+1;
            elseif reg_0(j,1)>=36526 && reg_0(j,1)<=36891

feederFAULT00(i+feederindex,44)=feederFAULT00(i+feederindex,44)+1;
```

```
            elseif reg_0(j,1)>=36892 && reg_0(j,1)<=37256

feederFAULT01(i+feederindex,44)=feederFAULT01(i+feederindex,44)+1;
            elseif reg_0(j,1)>=37257 && reg_0(j,1)<=37621

feederFAULT02(i+feederindex,44)=feederFAULT02(i+feederindex,44)+1;
            elseif reg_0(j,1)>=37622 && reg_0(j,1)<=37986

feederFAULT03(i+feederindex,44)=feederFAULT03(i+feederindex,44)+1;
            elseif reg_0(j,1)>=37987 && reg_0(j,1)<=38352

feederFAULT04(i+feederindex,44)=feederFAULT04(i+feederindex,44)+1;
            end
            feederFAULT(i+feederindex,44)=feederFAULT(i+feederindex,44)+1;
          end


       end

    end

    if feederoutputMEDDED(i+feederindex,4)==0
       feederoutputMEDDED(i+feederindex,8)=999;
       feederoutputMEDDED(i+feederindex,9)=999;
       feederoutputMEDDED(i+feederindex,10)=999;
       feederoutputMEDDED(i+feederindex,11)=999;
       feederoutputMEDDED(i+feederindex,12)=999;
       feederoutputMEDDED(i+feederindex,13)=999;
       feederoutputMEDDED(i+feederindex,14)=999;
    end




 end



%*****************************************************************
****
  %----SAIFI W/O MED CALCULATION-----------------------------------------
```

```
%*********************************************************************
****

   for i=1:feedernames_size(1,1)

      if i<feedernames_size(1,1)
         for j=feedernames(i,2):(feedernames(i+1,2)-1)

            % SAIFI----------use affected clients(row 12 reg_0) and total clients (row 3
feederclients)
            %Saifi will be calculated with first instance of an interruption. If
            %sectionalization occurs, this subsequent smaller interruptions will not be
            %used for calculation purposes.
            if (j==1)&&(reg_0(j,7)>momentary)

               if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
                  if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                     clients=reg_0(j,12);
                  else
                     clients=feederoutputMEDDED(i+feederindex,4);
                  end
                  if clients==0
                     clients=99;
                  end
                  saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

                  if (saididaily<feederMEDtuki(i+feederindex,4))&&(r~=4)&&(clients>50)

feederoutputMEDDED(i+feederindex,16)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,16);

regdisoutputMEDDED(i+feederindex,16)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,16);
                  end

feederoutput(i+feederindex,16)=(reg_0(j,12))/clients+feederoutput(i+feederindex,16);

regdisoutput(i+feederindex,16)=(reg_0(j,12))+regdisoutput(i+feederindex,16);


               end
               if reg_0(j,1)>=36161 && reg_0(j,1)<=36525
                  if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                     clients=reg_0(j,12);
```

```
                else
                   clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                   clients=99;
                end
                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

                if saididaily<feederMEDtuki(i+feederindex,5)&&(clients>50)

feederoutputMEDDED(i+feederindex,17)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,17);

regdisoutputMEDDED(i+feederindex,17)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,17);
                end

feederoutput(i+feederindex,17)=(reg_0(j,12))/clients+feederoutput(i+feederindex,17);

regdisoutput(i+feederindex,17)=(reg_0(j,12))+regdisoutput(i+feederindex,17);

             end
             if reg_0(j,1)>=36526 && reg_0(j,1)<=36891
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                   clients=reg_0(j,12);
                else
                   clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                   clients=99;
                end
                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

                if saididaily<feederMEDtuki(i+feederindex,6)&&(clients>50)

feederoutputMEDDED(i+feederindex,18)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,18);

regdisoutputMEDDED(i+feederindex,18)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,18);
                end

feederoutput(i+feederindex,18)=(reg_0(j,12))/clients+feederoutput(i+feederindex,18);

regdisoutput(i+feederindex,18)=(reg_0(j,12))+regdisoutput(i+feederindex,18);
```

```
                end
                if reg_0(j,1)>=36892 && reg_0(j,1)<=37256
                   if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                      clients=reg_0(j,12);
                   else
                      clients=feederoutputMEDDED(i+feederindex,4);
                   end
                   if clients==0
                      clients=99;
                   end
                   saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

                   if saididaily<feederMEDtuki(i+feederindex,7)&&(clients>50)

feederoutputMEDDED(i+feederindex,19)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,19);

regdisoutputMEDDED(i+feederindex,19)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,19);
                   end

feederoutput(i+feederindex,19)=(reg_0(j,12))/clients+feederoutput(i+feederindex,19);

regdisoutput(i+feederindex,19)=(reg_0(j,12))+regdisoutput(i+feederindex,19);

                end
                if reg_0(j,1)>=37257 && reg_0(j,1)<=37621
                   if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                      clients=reg_0(j,12);
                   else
                      clients=feederoutputMEDDED(i+feederindex,4);
                   end
                   if clients==0
                      clients=99;
                   end
                   saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

                   if saididaily<feederMEDtuki(i+feederindex,8)&&(clients>50)

feederoutputMEDDED(i+feederindex,20)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,20);

regdisoutputMEDDED(i+feederindex,20)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,20);
```

```
                end

feederoutput(i+feederindex,20)=(reg_0(j,12))/clients+feederoutput(i+feederindex,20);

regdisoutput(i+feederindex,20)=(reg_0(j,12))+regdisoutput(i+feederindex,20);

            end
            if reg_0(j,1)>=37622 && reg_0(j,1)<=37986
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end
               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

               if saididaily<feederMEDtuki(i+feederindex,9)&&(clients>50)

feederoutputMEDDED(i+feederindex,21)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,21);

regdisoutputMEDDED(i+feederindex,21)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,21);
                end

feederoutput(i+feederindex,21)=(reg_0(j,12))/clients+feederoutput(i+feederindex,21);

regdisoutput(i+feederindex,21)=(reg_0(j,12))+regdisoutput(i+feederindex,21);

            end
            if reg_0(j,1)>=37987 && reg_0(j,1)<=38352
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end
               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

               if saididaily<feederMEDtuki(i+feederindex,10)&&(clients>50)
```

```
feederoutputMEDDED(i+feederindex,22)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,22);

regdisoutputMEDDED(i+feederindex,22)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,22);
                    end

feederoutput(i+feederindex,22)=(reg_0(j,12))/clients+feederoutput(i+feederindex,22);

regdisoutput(i+feederindex,22)=(reg_0(j,12))+regdisoutput(i+feederindex,22);

                end
            end

            if j>1
                if ((reg_0(j,7)>momentary)&&(reg_0(j,2)~=reg_0(j-1,2)))
                    if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
                        if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                            clients=reg_0(j,12);
                        else
                            clients=feederoutputMEDDED(i+feederindex,4);
                        end
                        if clients==0
                            clients=99;
                        end
                        saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

                        if
(saididaily<feederMEDtuki(i+feederindex,4))&&(r~=4)&&(clients>50)

feederoutputMEDDED(i+feederindex,16)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,16);

regdisoutputMEDDED(i+feederindex,16)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,16);
                    end

feederoutput(i+feederindex,16)=(reg_0(j,12))/clients+feederoutput(i+feederindex,16);

regdisoutput(i+feederindex,16)=(reg_0(j,12))+regdisoutput(i+feederindex,16);

                    end
                    if reg_0(j,1)>=36161 && reg_0(j,1)<=36525
                        if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
```

```
                    clients=reg_0(j,12);
                else
                    clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                    clients=99;
                end
                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

                if saididaily<feederMEDtuki(i+feederindex,5)&&(clients>50)

feederoutputMEDDED(i+feederindex,17)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,17);

regdisoutputMEDDED(i+feederindex,17)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,17);
                end

feederoutput(i+feederindex,17)=(reg_0(j,12))/clients+feederoutput(i+feederindex,17);

regdisoutput(i+feederindex,17)=(reg_0(j,12))+regdisoutput(i+feederindex,17);

            end
            if reg_0(j,1)>=36526 && reg_0(j,1)<=36891
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                    clients=reg_0(j,12);
                else
                    clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                    clients=99;
                end
                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

                if saididaily<feederMEDtuki(i+feederindex,6)&&(clients>50)

feederoutputMEDDED(i+feederindex,18)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,18);

regdisoutputMEDDED(i+feederindex,18)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,18);
                end

feederoutput(i+feederindex,18)=(reg_0(j,12))/clients+feederoutput(i+feederindex,18);
```

```
regdisoutput(i+feederindex,18)=(reg_0(j,12))+regdisoutput(i+feederindex,18);

                end
                if reg_0(j,1)>=36892 && reg_0(j,1)<=37256
                   if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                      clients=reg_0(j,12);
                   else
                      clients=feederoutputMEDDED(i+feederindex,4);
                   end
                   if clients==0
                      clients=99;
                   end
                   saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

                   if saididaily<feederMEDtuki(i+feederindex,7)&&(clients>50)

feederoutputMEDDED(i+feederindex,19)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,19);

regdisoutputMEDDED(i+feederindex,19)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,19);
                   end

feederoutput(i+feederindex,19)=(reg_0(j,12))/clients+feederoutput(i+feederindex,19);

regdisoutput(i+feederindex,19)=(reg_0(j,12))+regdisoutput(i+feederindex,19);

                end
                if reg_0(j,1)>=37257 && reg_0(j,1)<=37621
                   if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                      clients=reg_0(j,12);
                   else
                      clients=feederoutputMEDDED(i+feederindex,4);
                   end
                   if clients==0
                      clients=99;
                   end
                   saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

                   if saididaily<feederMEDtuki(i+feederindex,8)&&(clients>50)

feederoutputMEDDED(i+feederindex,20)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,20);
```

```
regdisoutputMEDDED(i+feederindex,20)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,20);
                    end

feederoutput(i+feederindex,20)=(reg_0(j,12))/clients+feederoutput(i+feederindex,20);

regdisoutput(i+feederindex,20)=(reg_0(j,12))+regdisoutput(i+feederindex,20);

                end
                if reg_0(j,1)>=37622 && reg_0(j,1)<=37986
                   if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                      clients=reg_0(j,12);
                   else
                      clients=feederoutputMEDDED(i+feederindex,4);
                   end
                   if clients==0
                      clients=99;
                   end
                   saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

                   if saididaily<feederMEDtuki(i+feederindex,9)&&(clients>50)

feederoutputMEDDED(i+feederindex,21)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,21);

regdisoutputMEDDED(i+feederindex,21)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,21);
                    end

feederoutput(i+feederindex,21)=(reg_0(j,12))/clients+feederoutput(i+feederindex,21);

regdisoutput(i+feederindex,21)=(reg_0(j,12))+regdisoutput(i+feederindex,21);

                end
                if reg_0(j,1)>=37987 && reg_0(j,1)<=38352
                   if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                      clients=reg_0(j,12);
                   else
                      clients=feederoutputMEDDED(i+feederindex,4);
                   end
                   if clients==0
                      clients=99;
                   end
                   saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
```

```
                    if saididaily<feederMEDtuki(i+feederindex,10)&&(clients>50)

feederoutputMEDDED(i+feederindex,22)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,22);

regdisoutputMEDDED(i+feederindex,22)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,22);
                    end

feederoutput(i+feederindex,22)=(reg_0(j,12))/clients+feederoutput(i+feederindex,22);

regdisoutput(i+feederindex,22)=(reg_0(j,12))+regdisoutput(i+feederindex,22);

                end

              end
            end

          end

        end
        if i==feedernames_size(1,1) %If added to fix the problem when the last feeder was
read on feedernames, which had no next feeder to use as end index in the previous IF.

          for j=feedernames(feedernames_size(1,1),2):reg_0_size(1,1)

            if ((reg_0(j,7)>momentary)&&(reg_0(j,2)~=reg_0(j-1,2)))

              if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
                else
                  clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                  clients=99;
                end
                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

                if (saididaily<feederMEDtuki(i+feederindex,4))&&(r~=4)&&(clients>50)

feederoutputMEDDED(i+feederindex,16)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,16);
```

```
regdisoutputMEDDED(i+feederindex,16)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,16);
                end

feederoutput(i+feederindex,16)=(reg_0(j,12))/clients+feederoutput(i+feederindex,16);

regdisoutput(i+feederindex,16)=(reg_0(j,12))+regdisoutput(i+feederindex,16);
            end

            if reg_0(j,1)>=36161 && reg_0(j,1)<=36525
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end
               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

               if saididaily<feederMEDtuki(i+feederindex,5)&&(clients>50)

feederoutputMEDDED(i+feederindex,17)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,17);

regdisoutputMEDDED(i+feederindex,17)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,17);
                end

feederoutput(i+feederindex,17)=(reg_0(j,12))/clients+feederoutput(i+feederindex,17);

regdisoutput(i+feederindex,17)=(reg_0(j,12))+regdisoutput(i+feederindex,17);
            end

            if reg_0(j,1)>=36526 && reg_0(j,1)<=36891
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end
               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
```

```
                if saididaily<feederMEDtuki(i+feederindex,6)&&(clients>50)

feederoutputMEDDED(i+feederindex,18)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,18);

regdisoutputMEDDED(i+feederindex,18)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,18);
                end

feederoutput(i+feederindex,18)=(reg_0(j,12))/clients+feederoutput(i+feederindex,18);

regdisoutput(i+feederindex,18)=(reg_0(j,12))+regdisoutput(i+feederindex,18);
            end

            if reg_0(j,1)>=36892 && reg_0(j,1)<=37256
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end
               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

               if saididaily<feederMEDtuki(i+feederindex,7)&&(clients>50)

feederoutputMEDDED(i+feederindex,19)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,19);

regdisoutputMEDDED(i+feederindex,19)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,19);
                end

feederoutput(i+feederindex,19)=(reg_0(j,12))/clients+feederoutput(i+feederindex,19);

regdisoutput(i+feederindex,19)=(reg_0(j,12))+regdisoutput(i+feederindex,19);
            end

            if reg_0(j,1)>=37257 && reg_0(j,1)<=37621
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
```

```
                end
                if clients==0
                   clients=99;
                end
                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

                if saididaily<feederMEDtuki(i+feederindex,8)&&(clients>50)

feederoutputMEDDED(i+feederindex,20)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,20);

regdisoutputMEDDED(i+feederindex,20)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,20);
                end

feederoutput(i+feederindex,20)=(reg_0(j,12))/clients+feederoutput(i+feederindex,20);

regdisoutput(i+feederindex,20)=(reg_0(j,12))+regdisoutput(i+feederindex,20);
            end

            if reg_0(j,1)>=37622 && reg_0(j,1)<=37986
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                   clients=reg_0(j,12);
                else
                   clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                   clients=99;
                end
                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

                if saididaily<feederMEDtuki(i+feederindex,9)&&(clients>50)

feederoutputMEDDED(i+feederindex,21)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,21);

regdisoutputMEDDED(i+feederindex,21)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,21);
                end

feederoutput(i+feederindex,21)=(reg_0(j,12))/clients+feederoutput(i+feederindex,21);

regdisoutput(i+feederindex,21)=(reg_0(j,12))+regdisoutput(i+feederindex,21);

            end
```

```
            if reg_0(j,1)>=37987 && reg_0(j,1)<=38352
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
               end
               if clients==0
                  clients=99;
                  endsaididaily=(reg_0(j,12)*reg_0(j,7))/clients;
                  if saididaily<feederMEDtuki(i+feederindex,10)&&(clients>50)

feederoutputMEDDED(i+feederindex,22)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,22);

regdisoutputMEDDED(i+feederindex,22)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,22);
                  end

feederoutput(i+feederindex,22)=(reg_0(j,12))/clients+feederoutput(i+feederindex,22);

regdisoutput(i+feederindex,22)=(reg_0(j,12))+regdisoutput(i+feederindex,22);
               end
            end

         end
      end

   end
   if feederoutputMEDDED(i+feederindex,4)==0
      feederoutputMEDDED(i+feederindex,16)=999;
      feederoutputMEDDED(i+feederindex,17)=999;
      feederoutputMEDDED(i+feederindex,18)=999;
      feederoutputMEDDED(i+feederindex,19)=999;
      feederoutputMEDDED(i+feederindex,20)=999;
      feederoutputMEDDED(i+feederindex,21)=999;
      feederoutputMEDDED(i+feederindex,22)=999;
      feederoutput(i+feederindex,16)=999;
      feederoutput(i+feederindex,17)=999;
      feederoutput(i+feederindex,18)=999;
      feederoutput(i+feederindex,19)=999;
      feederoutput(i+feederindex,20)=999;
      feederoutput(i+feederindex,21)=999;
      feederoutput(i+feederindex,22)=999;
   end
```

```
    end

    feederoutputMEDDED_size=size(feederoutputMEDDED);
    feederindex=feederindex+feedernames_size(1,1);

    if r==0
       reg0_saididaily=saididailytemp;
       reg0_saididailyMEDDED=saididailytempMEDDED;
    end
    if r==1
       reg1_saididaily=saididailytemp;
       reg1_saididailyMEDDED=saididailytempMEDDED;
    end
    if r==2
       reg2_saididaily=saididailytemp;
       reg2_saididailyMEDDED=saididailytempMEDDED;
    end
    if r==3
       reg3_saididaily=saididailytemp;
       reg3_saididailyMEDDED=saididailytempMEDDED;
    end
    if r==4
       reg4_saididaily=saididailytemp;
       reg4_saididailyMEDDED=saididailytempMEDDED;
    end
    if r==5
       reg5_saididaily=saididailytemp;
       reg5_saididailyMEDDED=saididailytempMEDDED;
    end
    if r==6
       reg6_saididaily=saididailytemp;
       reg6_saididailyMEDDED=saididailytempMEDDED;
    end

end



%-------------------------------------------------------------------------%
%//////////////////////END OF MEDDED SAIDI SAIFI//////////////////////////%
%-------------------------------------------------------------------------%

for i=1:3
    feederFAULT(:,1)=[];
    feederFAULT98(:,1)=[];
```

```
    feederFAULT99(:,1)=[];
    feederFAULT00(:,1)=[];
    feederFAULT01(:,1)=[];
    feederFAULT02(:,1)=[];
    feederFAULT03(:,1)=[];
    feederFAULT04(:,1)=[];

end



feederFAULT=horzcat(feederFAULT_append,feederFAULT);
feederFAULT98=horzcat(feederFAULT_append,feederFAULT98);
feederFAULT99=horzcat(feederFAULT_append,feederFAULT99);
feederFAULT00=horzcat(feederFAULT_append,feederFAULT00);
feederFAULT01=horzcat(feederFAULT_append,feederFAULT01);
feederFAULT02=horzcat(feederFAULT_append,feederFAULT02);
feederFAULT03=horzcat(feederFAULT_append,feederFAULT03);
feederFAULT04=horzcat(feederFAULT_append,feederFAULT04);

%-------------------------------------------------------------------------
%//////////////////////////New Cleaner Module//////////////////////////////
%-------------------------------------------------------------------------

%This new module will only kill client-less feeders, thus prevwnting the
%999 issue.

killed=0;
counter=1;



feederFAULTcodes=[13,38,39,48,51,52,53,54,56,58,59,63,65,66,67,69,83,85,86,87,88,8
9,90,91,92,93,94,95,96,97,98,99];

feederFAULT_size=size(feederFAULT);

for j=1:8


    for i=1:feederFAULT_size(1,1)

        if j==1
            row=feederFAULT(i,:);
        end
```

```
if j==2
   row=feederFAULT98(i,:);
end
if j==3
   row=feederFAULT99(i,:);
end
if j==4
   row=feederFAULT00(i,:);
end
if j==5
   row=feederFAULT01(i,:);
end
if j==6
   row=feederFAULT02(i,:);
end
if j==7
   row=feederFAULT03(i,:);
end
if j==8
   row=feederFAULT04(i,:);
end

row(1,37)=0;
row(1,35)=0;
row(:,44)=0;
row(:,43)=0;
row(:,42)=0;
row(:,41)=0;
row(:,40)=0;
row(:,39)=0;
row(:,38)=0;
row(:,1)=[];
row(:,1)=[];
row(:,1)=[];

[MAXI,x]=max(row);

if j==1
   feederFAULT(i,37)=feederFAULTcodes(1,x);
end
if j==2
   feederFAULT98(i,37)=feederFAULTcodes(1,x);
end
if j==3
   feederFAULT99(i,37)=feederFAULTcodes(1,x);
```

```
        end
        if j==4
            feederFAULT00(i,37)=feederFAULTcodes(1,x);
        end
        if j==5
            feederFAULT01(i,37)=feederFAULTcodes(1,x);
        end
        if j==6
            feederFAULT02(i,37)=feederFAULTcodes(1,x);
        end
        if j==7
            feederFAULT03(i,37)=feederFAULTcodes(1,x);
        end
        if j==8
            feederFAULT04(i,37)=feederFAULTcodes(1,x);
        end

    end


end

save feederFAULTS.mat feederFAULT*

districtFAULT=zeros(27,44); %FORMAT: 1-Region 2-District 3-ZERO 4-10-SAIDI 11-
ZERO 12-19-SAIFI
districtFAULT98=zeros(27,44);
districtFAULT99=zeros(27,44);
districtFAULT00=zeros(27,44);
districtFAULT01=zeros(27,44);
districtFAULT02=zeros(27,44);
districtFAULT03=zeros(27,44);
districtFAULT04=zeros(27,44);
districtFAULTTEMP=zeros(27,44);



for x=1:8
    index=1;

    if x==1
        feederFAULTTEMP=feederFAULT;
    elseif x==2
        feederFAULTTEMP=feederFAULT98;
    elseif x==3
```

```
      feederFAULTTEMP=feederFAULT99;
   elseif x==4
      feederFAULTTEMP=feederFAULT00;
   elseif x==5
      feederFAULTTEMP=feederFAULT01;
   elseif x==6
      feederFAULTTEMP=feederFAULT02;
   elseif x==7
      feederFAULTTEMP=feederFAULT03;
   elseif x==8
      feederFAULTTEMP=feederFAULT04;
   end
   districtFAULTTEMP=zeros(27,44);
   for
disnum=[20,22,23,24,95,10,11,12,13,14,34,40,42,44,45,50,53,54,55,60,61,62,63,70,71,7
2,73]

      for i=1:feederFAULT_size(1,1)
         districtFAULTTEMP(index,2)=disnum;
         if feederFAULTTEMP(i,2)==disnum

            districtFAULTTEMP(index,1)=feederFAULTTEMP(i,1);

            for j=4:44

districtFAULTTEMP(index,j)=feederFAULTTEMP(i,j)+districtFAULTTEMP(index,j);
            end
         end
      end
      index=index+1;
   end

   districtFAULTTEMP(1,1)=0;
   districtFAULTTEMP(2,1)=0;
   districtFAULTTEMP(3,1)=0;
   districtFAULTTEMP(4,1)=0;
   districtFAULTTEMP(5,1)=0;
   districtFAULTTEMP(6,1)=1;
   districtFAULTTEMP(7,1)=1;
   districtFAULTTEMP(8,1)=1;
   districtFAULTTEMP(9,1)=2;
   districtFAULTTEMP(10,1)=2;
   districtFAULTTEMP(11,1)=2;
   districtFAULTTEMP(12,1)=3;
   districtFAULTTEMP(13,1)=3;
```

```
districtFAULTTEMP(14,1)=3;
districtFAULTTEMP(15,1)=3;
districtFAULTTEMP(16,1)=4;
districtFAULTTEMP(17,1)=4;
districtFAULTTEMP(18,1)=4;
districtFAULTTEMP(19,1)=4;
districtFAULTTEMP(20,1)=5;
districtFAULTTEMP(21,1)=5;
districtFAULTTEMP(22,1)=5;
districtFAULTTEMP(23,1)=5;
districtFAULTTEMP(24,1)=6;
districtFAULTTEMP(25,1)=6;
districtFAULTTEMP(26,1)=6;
districtFAULTTEMP(27,1)=6;

for y=1:27

   row=districtFAULTTEMP(y,:);
   row(1,37)=0;
   row(1,35)=0;
   row(:,43)=0;
   row(:,42)=0;
   row(:,41)=0;
   row(:,40)=0;
   row(:,39)=0;
   row(:,38)=0;
   row(:,1)=[];
   row(:,1)=[];
   row(:,1)=[];


   [MAXI,z]=max(row);
   districtFAULTTEMP(y,37)=feederFAULTcodes(1,z);

end

if x==1
   districtFAULT=districtFAULTTEMP;
elseif x==2
   districtFAULT98=districtFAULTTEMP;
elseif x==3
   districtFAULT99=districtFAULTTEMP;
elseif x==4
   districtFAULT00=districtFAULTTEMP;
elseif x==5
```

```
      districtFAULT01=districtFAULTTEMP;
   elseif x==6
      districtFAULT02=districtFAULTTEMP;
   elseif x==7
      districtFAULT03=districtFAULTTEMP;
   elseif x==8
      districtFAULT04=districtFAULTTEMP;
   end

end




save outputs.mat feederoutput feederoutputMEDDED regdisoutput
regdisoutputMEDDED

for k=1:(feederoutput_size)

   if feederoutput(counter,4)==0
      feederoutput(counter,:)=[];
      feederMEDtuki(counter,:)=[];
      feederoutputMEDDED(counter,:)=[];
      feederFAULT(counter,:)=[];
      feederFAULT98(counter,:)=[];
      feederFAULT99(counter,:)=[];
      feederFAULT00(counter,:)=[];
      feederFAULT01(counter,:)=[];
      feederFAULT02(counter,:)=[];
      feederFAULT03(counter,:)=[];
      feederFAULT04(counter,:)=[];
      regdisoutput(counter,:)=[];
      regdisoutputMEDDED(counter,:)=[];
      killed=killed+1;
      counter=counter-1;
   end
   counter=counter+1;
end
feederoutput_size=size(feederoutput);
feederMEDtuki_size=size(feederMEDtuki);
feederoutputMEDDED_size=size(feederoutputMEDDED);
feederFAULT_size=size(feederFAULT);

%--------------------------------------------------------------------------
```

```
%//////////////////////////CONGLOMERATOR////////////////////////////////
%-------------------------------------------------------------------------

systemoutput=zeros(1,17);


%----Conglomerate by Region

regionoutput=zeros(7,17);
regionoutputMEDDED=zeros(7,17);


index=1;
for regnum=[0,1,2,3,4,5,6]
   for i=1:feederoutput_size(1,1)

      if feederoutput(i,1)==regnum
         regionoutput(index,1)=regnum;

         %SAIDI
         regionoutput(index,3)=regionoutput(index,3)+regdisoutput(i,8); %98
         regionoutput(index,4)=regionoutput(index,4)+regdisoutput(i,9); %99
         regionoutput(index,5)=regionoutput(index,5)+regdisoutput(i,10); %00
         regionoutput(index,6)=regionoutput(index,6)+regdisoutput(i,11); %01
         regionoutput(index,7)=regionoutput(index,7)+regdisoutput(i,12); %02
         regionoutput(index,8)=regionoutput(index,8)+regdisoutput(i,13); %03
         regionoutput(index,9)=regionoutput(index,9)+regdisoutput(i,14); %04

         %SAIFI
         regionoutput(index,11)=regionoutput(index,11)+regdisoutput(i,16); %98
         regionoutput(index,12)=regionoutput(index,12)+regdisoutput(i,17); %99
         regionoutput(index,13)=regionoutput(index,13)+regdisoutput(i,18); %00
         regionoutput(index,14)=regionoutput(index,14)+regdisoutput(i,19); %01
         regionoutput(index,15)=regionoutput(index,15)+regdisoutput(i,20); %02
         regionoutput(index,16)=regionoutput(index,16)+regdisoutput(i,21); %03
         regionoutput(index,17)=regionoutput(index,17)+regdisoutput(i,22); %04


         %SAIDI MEDDED

regionoutputMEDDED(index,3)=regionoutputMEDDED(index,3)+regdisoutputMEDDE
D(i,8); %98

regionoutputMEDDED(index,4)=regionoutputMEDDED(index,4)+regdisoutputMEDDE
D(i,9); %99
```

```
regionoutputMEDDED(index,5)=regionoutputMEDDED(index,5)+regdisoutputMEDDE
D(i,10); %00

regionoutputMEDDED(index,6)=regionoutputMEDDED(index,6)+regdisoutputMEDDE
D(i,11); %01

regionoutputMEDDED(index,7)=regionoutputMEDDED(index,7)+regdisoutputMEDDE
D(i,12); %02

regionoutputMEDDED(index,8)=regionoutputMEDDED(index,8)+regdisoutputMEDDE
D(i,13); %03

regionoutputMEDDED(index,9)=regionoutputMEDDED(index,9)+regdisoutputMEDDE
D(i,14); %04


        %SAIFI MEDDED

regionoutputMEDDED(index,11)=regionoutputMEDDED(index,11)+regdisoutputMED
DED(i,16); %98

regionoutputMEDDED(index,12)=regionoutputMEDDED(index,12)+regdisoutputMED
DED(i,17); %99

regionoutputMEDDED(index,13)=regionoutputMEDDED(index,13)+regdisoutputMED
DED(i,18); %00

regionoutputMEDDED(index,14)=regionoutputMEDDED(index,14)+regdisoutputMED
DED(i,19); %01

regionoutputMEDDED(index,15)=regionoutputMEDDED(index,15)+regdisoutputMED
DED(i,20); %02

regionoutputMEDDED(index,16)=regionoutputMEDDED(index,16)+regdisoutputMED
DED(i,21); %03

regionoutputMEDDED(index,17)=regionoutputMEDDED(index,17)+regdisoutputMED
DED(i,22); %04


    end

  end
```

```
    index=index+1;


end

systemoutput=(regionoutput(1,:)+regionoutput(2,:)+regionoutput(3,:)+regionoutput(4,:)+
regionoutput(5,:)+regionoutput(6,:)+regionoutput(7,:))/systemclients;
systemoutputMEDDED=(regionoutputMEDDED(1,:)+regionoutputMEDDED(2,:)+regio
noutputMEDDED(3,:)+regionoutputMEDDED(4,:)+regionoutputMEDDED(5,:)+regiono
utputMEDDED(6,:)+regionoutputMEDDED(7,:))/systemclients;

tukitu=regionoutput;
tukituMEDDED=regionoutputMEDDED;

regionoutput(1,:)=regionoutput(1,:)/regionclients(1,2);
regionoutput(2,:)=regionoutput(2,:)/regionclients(2,2);
regionoutput(3,:)=regionoutput(3,:)/regionclients(3,2);
regionoutput(4,:)=regionoutput(4,:)/regionclients(4,2);
regionoutput(5,:)=regionoutput(5,:)/regionclients(5,2);
regionoutput(6,:)=regionoutput(6,:)/regionclients(6,2);
regionoutput(7,:)=regionoutput(7,:)/regionclients(7,2);

regionoutputMEDDED(1,:)=regionoutputMEDDED(1,:)/regionclients(1,2);
regionoutputMEDDED(2,:)=regionoutputMEDDED(2,:)/regionclients(2,2);
regionoutputMEDDED(3,:)=regionoutputMEDDED(3,:)/regionclients(3,2);
regionoutputMEDDED(4,:)=regionoutputMEDDED(4,:)/regionclients(4,2);
regionoutputMEDDED(5,:)=regionoutputMEDDED(5,:)/regionclients(5,2);
regionoutputMEDDED(6,:)=regionoutputMEDDED(6,:)/regionclients(6,2);
regionoutputMEDDED(7,:)=regionoutputMEDDED(7,:)/regionclients(7,2);

regionoutput(1,1)=0;
regionoutput(2,1)=1;
regionoutput(3,1)=2;
regionoutput(4,1)=3;
regionoutput(5,1)=4;
regionoutput(6,1)=5;
regionoutput(7,1)=6;

regionoutputMEDDED(1,1)=0;
regionoutputMEDDED(2,1)=1;
regionoutputMEDDED(3,1)=2;
regionoutputMEDDED(4,1)=3;
regionoutputMEDDED(5,1)=4;
regionoutputMEDDED(6,1)=5;
regionoutputMEDDED(7,1)=6;
```

```
%-----Conglomerate Data by District---------------------------------------

districtoutput=zeros(27,18); %FORMAT: 1-Region 2-District 3-ZERO 4-10-SAIDI 11-
ZERO 12-19-SAIFI
districtoutputMEDDED=zeros(27,18);
index=1;
for
disnum=[20,22,23,24,95,10,11,12,13,14,34,40,42,44,45,50,53,54,55,60,61,62,63,70,71,7
2,73]
   for i=1:feederoutput_size(1,1)

      if regdisoutput(i,2)==disnum

         %SAIDI

         districtoutput(index,2)=disnum;
         districtoutput(index,3)=districtoutput(index,3)+regdisoutput(i,8); %98
         districtoutput(index,4)=districtoutput(index,4)+regdisoutput(i,9); %99
         districtoutput(index,5)=districtoutput(index,5)+regdisoutput(i,10);%00
         districtoutput(index,6)=districtoutput(index,6)+regdisoutput(i,11);%01
         districtoutput(index,7)=districtoutput(index,7)+regdisoutput(i,12);%02
         districtoutput(index,8)=districtoutput(index,8)+regdisoutput(i,13);%03
         districtoutput(index,9)=districtoutput(index,9)+regdisoutput(i,14);%04

         %SAIFI

         districtoutput(index,11)=districtoutput(index,11)+regdisoutput(i,16);%98
         districtoutput(index,12)=districtoutput(index,12)+regdisoutput(i,17);%99
         districtoutput(index,13)=districtoutput(index,13)+regdisoutput(i,18);%00
         districtoutput(index,14)=districtoutput(index,14)+regdisoutput(i,19);%01
         districtoutput(index,15)=districtoutput(index,15)+regdisoutput(i,20);%02
         districtoutput(index,16)=districtoutput(index,16)+regdisoutput(i,21);%03
         districtoutput(index,17)=districtoutput(index,17)+regdisoutput(i,22);%04

         %SAIDI MEDDED

         districtoutputMEDDED(index,2)=disnum;

districtoutputMEDDED(index,3)=districtoutputMEDDED(index,3)+regdisoutputMEDD
ED(i,8); %98
```

```
districtoutputMEDDED(index,4)=districtoutputMEDDED(index,4)+regdisoutputMEDD
ED(i,9); %99

districtoutputMEDDED(index,5)=districtoutputMEDDED(index,5)+regdisoutputMEDD
ED(i,10);%00

districtoutputMEDDED(index,6)=districtoutputMEDDED(index,6)+regdisoutputMEDD
ED(i,11);%01

districtoutputMEDDED(index,7)=districtoutputMEDDED(index,7)+regdisoutputMEDD
ED(i,12);%02

districtoutputMEDDED(index,8)=districtoutputMEDDED(index,8)+regdisoutputMEDD
ED(i,13);%03

districtoutputMEDDED(index,9)=districtoutputMEDDED(index,9)+regdisoutputMEDD
ED(i,14);%04

        %SAIFI MEDDED


districtoutputMEDDED(index,11)=districtoutputMEDDED(index,11)+regdisoutputMED
DED(i,16);%98

districtoutputMEDDED(index,12)=districtoutputMEDDED(index,12)+regdisoutputMED
DED(i,17);%99

districtoutputMEDDED(index,13)=districtoutputMEDDED(index,13)+regdisoutputMED
DED(i,18);%00

districtoutputMEDDED(index,14)=districtoutputMEDDED(index,14)+regdisoutputMED
DED(i,19);%01

districtoutputMEDDED(index,15)=districtoutputMEDDED(index,15)+regdisoutputMED
DED(i,20);%02

districtoutputMEDDED(index,16)=districtoutputMEDDED(index,16)+regdisoutputMED
DED(i,21);%03

districtoutputMEDDED(index,17)=districtoutputMEDDED(index,17)+regdisoutputMED
DED(i,22);%04

      end
```

```
   end
   index=index+1;


end

for i=1:27
   for j=3:18

      districtoutput(i,j)=districtoutput(i,j)/districtclients(i,2);

   end
end

for i=1:27
   for j=3:18

      districtoutputMEDDED(i,j)=districtoutputMEDDED(i,j)/districtclients(i,2);

   end
end



systemFAULT=zeros(1,35);

systemFAULT=sum(districtFAULT);

systemFAULT(1,1)=0;
systemFAULT(1,2)=0;

%feeder_list %Feederclients column one. used as index for for. other columns
eliminated.

for i=1:feederclients_size(1,1)

   for j=1:feederoutput_size(1,1)

      if feederclients(i,1)==feederoutput(j,3)

         feederclients_append(i,2)=feederclients_append(i,1)+feederoutput(j,8);
         feederclients_append(i,3)=feederclients_append(i,3)+feederoutput(j,9);
         feederclients_append(i,4)=feederclients_append(i,4)+feederoutput(j,10);
         feederclients_append(i,5)=feederclients_append(i,5)+feederoutput(j,11);
         feederclients_append(i,6)=feederclients_append(i,6)+feederoutput(j,12);
```

```
feederclients_append(i,7)=feederclients_append(i,7)+feederoutput(j,13);
feederclients_append(i,8)=feederclients_append(i,8)+feederoutput(j,14);

feederclients_append(i,10)=feederclients_append(i,10)+feederoutput(j,16);
feederclients_append(i,11)=feederclients_append(i,11)+feederoutput(j,17);
feederclients_append(i,12)=feederclients_append(i,12)+feederoutput(j,18);
feederclients_append(i,13)=feederclients_append(i,13)+feederoutput(j,19);
feederclients_append(i,14)=feederclients_append(i,14)+feederoutput(j,20);
feederclients_append(i,15)=feederclients_append(i,15)+feederoutput(j,21);
feederclients_append(i,16)=feederclients_append(i,16)+feederoutput(j,22);


feederclients_append_MEDDED(i,2)=feederclients_append_MEDDED(i,2)+feederoutpu
t(j,8);

feederclients_append_MEDDED(i,3)=feederclients_append_MEDDED(i,3)+feederoutpu
t(j,9);

feederclients_append_MEDDED(i,4)=feederclients_append_MEDDED(i,4)+feederoutpu
t(j,10);

feederclients_append_MEDDED(i,5)=feederclients_append_MEDDED(i,5)+feederoutpu
t(j,11);

feederclients_append_MEDDED(i,6)=feederclients_append_MEDDED(i,6)+feederoutpu
t(j,12);

feederclients_append_MEDDED(i,7)=feederclients_append_MEDDED(i,7)+feederoutpu
t(j,13);

feederclients_append_MEDDED(i,8)=feederclients_append_MEDDED(i,8)+feederoutpu
t(j,14);


feederclients_append_MEDDED(i,10)=feederclients_append_MEDDED(i,10)+feederout
put(j,16);

feederclients_append_MEDDED(i,11)=feederclients_append_MEDDED(i,11)+feederout
put(j,17);

feederclients_append_MEDDED(i,12)=feederclients_append_MEDDED(i,12)+feederout
put(j,18);

feederclients_append_MEDDED(i,13)=feederclients_append_MEDDED(i,13)+feederout
put(j,19);
```

```
feederclients_append_MEDDED(i,14)=feederclients_append_MEDDED(i,14)+feederoutput(j,20);

feederclients_append_MEDDED(i,15)=feederclients_append_MEDDED(i,15)+feederoutput(j,21);

feederclients_append_MEDDED(i,16)=feederclients_append_MEDDED(i,16)+feederoutput(j,22);

      end
   end
end

feederclients_total=horzcat(feederclients,feederclients_append);
feederclients_total_MEDDED=horzcat(feederclients,feederclients_append_MEDDED);

save saididailys.mat *saididaily *saididailyMEDDED

save feederoutput.txt feederoutput -ascii
save feederoutputMEDDED.txt feederoutputMEDDED -ascii
save regionoutput.txt regionoutput -ascii
save regionoutputMEDDED.txt regionoutputMEDDED -ascii
save districtoutput.txt districtoutput -ascii
save districtoutputMEDDED.txt districtoutputMEDDED -ascii



toc
```

# APPENDIX B: STRIPPER SCRIPT

```
%Matlab M file for breakingcodeappart Excel Files derived from the  dBASE
%database. It breaks appart interruption data files for each district (27)
%and makes smaller files for each feeder (1172). During the process it
%should filter bad data using the additional columns of Date and TIme check
%in the excell file. The excell file is made with the dbase file with
%several modifications to make it compatible with matlab's xlsread command.

%-------------------------------------------------------------------------%
%////////////////////////////START /////////////////////////////////////--1-%
%-------------------------------------------------------------------------%
clear all
tic
load regiones.mat
load weights.mat
momentary=5; %Threshold value (min) for momentary events
feedernames=zeros(2,3); %Init of Feedernames. Contains feedernumber and feederclients
for each region. It is reassigned for each region iteration.
systemclients=1537306; %Value for total system clients
feedernames_size=size(feedernames); %No desc. necessary
feederoutput=zeros(feedernames_size(1,1),23);%Output file. See format in ripper.xls
feederoutputMED=zeros(feedernames_size(1,1),15);%Output file for 97 SAIDI section.
See format in ripper.xls
regdisoutput=zeros(feedernames_size(1,1),23);%Output file. Results NOT divided by
feeder clients. Used for region/district calcs.
regdisoutputMEDDED=zeros(feedernames_size(1,1),23);%Output file. MED removed.
Results NOT divided by feeder clients. Used for region/district calcs.


feederclients_size=size(feederclients);%Size of feederclients. Feederclients is loaded
from regiones.mat. It contains the master list for feeder#, clients, region and district

feederclients_append=zeros(feederclients_size(1,1),16); %Append for final feeder data
output file. Will contain data from feederoutput, that will be appended to the 4 columns
from feederclients
feederclients_append_MEDDED=zeros(feederclients_size(1,1),16);

feederindex=0; %Feedernames offset value for each region iteration


feederMED=zeros(1,2); %Temporary file for MED data storage
feederMEDtuki=zeros(1,10); %Stores MED thresholds for each feeder/year.
```

```
MEDcount98=0;
MEDcount99=0;
MEDcount00=0;
MEDcount01=0;
MEDcount02=0;
MEDcount03=0;
MEDcount04=0;

SAIDItemp=zeros(1,1);
tempsize=size(region_0);
reg0_saididaily=zeros(tempsize(1,1),10);
reg0_saididailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_1);
reg1_saididaily=zeros(tempsize(1,1),10);
reg1_saididailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_2);
reg2_saididaily=zeros(tempsize(1,1),10);
reg2_saididailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_3);
reg3_saididaily=zeros(tempsize(1,1),10);
reg3_saididailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_4);
reg4_saididaily=zeros(tempsize(1,1),10);
reg4_saididailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_5);
reg5_saididaily=zeros(tempsize(1,1),10);
reg5_saididailyMEDDED=zeros(tempsize(1,1),10);
tempsize=size(region_6);
reg6_saididaily=zeros(tempsize(1,1),10);
reg6_saididailyMEDDED=zeros(tempsize(1,1),10);


%-------------------------------------------------------------------------%
%////////////////////////////START OF 97 MED//////////////////////////--2-%
%-------------------------------------------------------------------------%


% ------Feeders from the known good 98-04 FIles FDTB2 are used. Later a new
% r=0:6 for will open all 7 97FDT2 files to be used for MED SAIDI
% calculations. Region 4 has no 97 season, thus no
% MED values for year 1998 will be used.
%
% All regions except #4 have 97FDT2 files. Region #4 feeders are already part of the
% feederoutput. MED process will skip region #4 calculations.
```

```
inputoption=input('Adjustment Option 1-2-3-4 ');

feederindex=0;
for r=0:6
   if r==0

      reg_0=region_0_97; %Stores region 97 data in temp region varible.
      r
   end
   if r==1

      reg_0=region_1_97;
      r
   end
   if r==2

      reg_0=region_2_97;
      r
   end
   if r==3

      reg_0=region_3_97;
      r
   end
   if r==4

      %reg_0=region_4_97

   end
   if r==5

      reg_0=region_5_97;
      r
   end
   if r==6

      reg_0=region_6_97;
      r
   end

   reg_0_size=size(reg_0);
   feedernames=zeros(2,3);
   feedernames_size=size(feedernames);
```

```
   for i=1:reg_0_size(1,1)

      if i==1
         j=1;
         feedernames(j,1)=reg_0(i,11);
         feedernames(j,2)=1;
         feedernames(j,3)=reg_0(i,10);

      end

      if i>1 && reg_0(i,11)~=reg_0(i-1,11)
         j=j+1;

         feedernames(j,1)=reg_0(i,11);
         feedernames(j,2)=i; %Stores FIRST row where this feeder appears.
         feedernames(j,3)=reg_0(i,10);

      end
   end

   feedernames_size=size(feedernames);
   %Create a matrix to hold the feeder data (feederoutput). One row will be created for
each
   %feeder, the feeder name from variable feedernames will be included as
   %column one of this matrix. A for will be used from i=1->feedernames_size,
   %for filling this table with data. A nested for will rotate the reg_o
   %matrix looking for data from the feeder in the i row at the time and use
   %all its data to calculate SAIDI and SAIFI.

   for i=1:feedernames_size(1,1)
      feederoutputMED(i+feederindex,3)=feedernames(i,1);
      feederoutputMED(i+feederindex,2)=feedernames(i,3);

      for x=1:feederclients_size(1,1)

         if feederoutputMED(i+feederindex,3)==feederclients(x,1)
            feederoutputMED(i+feederindex,1)=feederclients(x,3); %Copy Region
            feederoutputMED(i+feederindex,4)=feederclients(x,2); %Copy Clients


            if feederoutputMED(i+feederindex,1)==feederclients(x,3)
               feederoutputMED(i+feederindex,2)=feederclients(x,4);      %Copy      Right
District
```

```
            end

        end

    end
end


%----------SAIDI 97 section.

for i=1:feedernames_size(1,1)


    if i<feedernames_size(1,1)
        MEDdatalog97=zeros((feedernames(i+1,2)-feedernames(i,2)),1);

        v=1;
        for j=feedernames(i,2):(feedernames(i+1,2)-1)

            % SAIDI-------------use clients(row 12) and duration (row 7)

            %feederoutput(i,1)=feedernames(i,1);
            if reg_0(j,7)>momentary %If prevents counting reclosings as part of SAIDI
                %Division by #clients will be moved inside this if
                %because: 1. Sum of ratios == a ratio of sums
                %2. THis way i can divide by the normal number of total
                %clients or those specified in the database, which
                %could be higher due to use of interties.

                if reg_0(j,12)>feederoutputMED(i+feederindex,4)
                    clients=reg_0(j,12);
                else
                    clients=feederoutputMED(i+feederindex,4);
                end
                if clients==0
                    clients=99;
                end
                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMED(i+feederindex,2),reg_0(j
,13),inputoption); %Adjustment step

feederoutputMED(i+feederindex,8)=saididaily+feederoutputMED(i+feederindex,8);
%SAIDI
```

```
            if saididaily~=0

               MEDdatalog97(v,1)=log(saididaily);

               v=v+1;
            end


         end


      end
      feederoutputMED(i+feederindex,9)=mean(MEDdatalog97);
      feederoutputMED(i+feederindex,10)=std(MEDdatalog97);

feederoutputMED(i+feederindex,11)=feederoutputMED(i+feederindex,9)+2.5*feederout
putMED(i+feederindex,10);
      feederoutputMED(i+feederindex,12)=exp(feederoutputMED(i+feederindex,11));
      MEDdatalog97old=MEDdatalog97;
      MEDdatalog97old_size=size(MEDdatalog97old);
      MEDdatalog97old_append=ones(MEDdatalog97old_size(1,1),1);

MEDdatalog97old_append=MEDdatalog97old_append*feederoutputMED(i+feederindex
,3);
      MEDdatalog97old=horzcat(MEDdatalog97old_append,MEDdatalog97old);
      feederMED=vertcat(feederMED,MEDdatalog97old);


   end
   if i==feedernames_size(1,1) %If added to fix the problem when the last feeder was
read on feedernames, which had no next feeder to use as end index in the previous IF.

      v=1;
      for j=feedernames(feedernames_size(1,1),2):reg_0_size(1,1)
         MEDdatalog97=zeros(reg_0_size(1,1)-feedernames(feedernames_size(1,1),1));

         % SAIDI----------use clients(row 12) and duration (row 7)
         if reg_0(j,7)>momentary %If prevents counting reclosings as part of SAIDI

            %Division by #clients will be moved inside this if
            %because: 1. Sum of ratios == a ratio of sums
            %2. THis way i can divide by the normal number of total
            %clients or those specified in the database, which
            %could be higher due to use of interties.
```

```
            if reg_0(j,12)>feederoutputMED(i+feederindex,4)
               clients=reg_0(j,12);
            else
               clients=feederoutputMED(i+feederindex,4);
            end
            if clients==0
               clients=99;
            end
            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMED(i+feederindex,2),reg_0(j
,13),inputoption); %Adjustment step

feederoutputMED(i+feederindex,8)=saididaily+feederoutputMED(i+feederindex,8);
%SAIDI

            if saididaily~=0

               MEDdatalog97(v,1)=log(saididaily);

               v=v+1;%Calculates and Stores daily SAIDI LOG
            end

            %New matrix created for storing the MEDdatalog97 for
            %each feeder, the feedernumber and the index location
            %at feederoutputMED. This index can be correlated with
            %that at feederoutput, via feedernames2.


         end


         v=v+1;

      end
      feederoutputMED(i+feederindex,9)=mean(MEDdatalog97);
      feederoutputMED(i+feederindex,10)=std(MEDdatalog97);

feederoutputMED(i+feederindex,11)=feederoutputMED(i+feederindex,9)+2.5*feederout
putMED(i+feederindex,10);
      feederoutputMED(i+feederindex,12)=exp(feederoutputMED(i+feederindex,11));
      MEDdatalog97old=MEDdatalog97;
      MEDdatalog97old_size=size(MEDdatalog97old);
      MEDdatalog97old_append=ones(MEDdatalog97old_size(1,1),1);
```

```
MEDdatalog97old_append=MEDdatalog97old_append*feederoutputMED(i+feederindex
,3);
          MEDdatalog97old=horzcat(MEDdatalog97old_append,MEDdatalog97old);
          feederMED=vertcat(feederMED,MEDdatalog97old);


       end

   end
   feederindex=feederindex+feedernames_size(1,1);
end
feederMED_size=size(feederMED);
%----FeederMED cleanup---------------------------------------
counter=1;
for k=1:(feederMED_size)

   if feederMED(counter,2)==0
      feederMED(counter,:)=[];
      counter=counter-1;
   end
   counter=counter+1;
end




%-------------------------------------------------------------------------%
%//////////////////////////////END OF 97 CALCS//////////////////////////--2-%
%-------------------------------------------------------------------------%

%-------------------------------------------------------------------------%
%//////////////////////////////START OF FEEDEROUPUT MED////////////////--3-%
%-------------------------------------------------------------------------%

feederindex=0;
for r=0:6
   if r==0
      reg_0=region_0;
      r

   end
   if r==1
      reg_0=region_1;
      r
```

```
end
if r==2
   reg_0=region_2;
   r

end
if r==3
   reg_0=region_3;
   r

end
if r==4
   reg_0=region_4;
   r

end
if r==5
   reg_0=region_5;
   r

end
if r==6
   reg_0=region_6;
   r

end

reg_0_size=size(reg_0);
feedernames=zeros(2,3);
feedernames_size=size(feedernames);

%---This for will read all the feeder column and extract the individual
%feeder names, which will be later used for sorting.

for i=1:reg_0_size(1,1)

   if i==1
      j=1;
      feedernames(j,1)=reg_0(i,11);
      feedernames(j,2)=1;
      feedernames(j,3)=reg_0(i,10);

   end
   if i>1 && reg_0(i,11)~=reg_0(i-1,11)
```

```
        j=j+1;
        feedernames(j,1)=reg_0(i,11);
        feedernames(j,2)=i; %Stores FIRST row where this feeder appears.
        feedernames(j,3)=reg_0(i,10);


      end



   end

   feedernames_size=size(feedernames);


   %Create a matrix to hold the feeder data (feederoutput). One row will be created for
each
   %feeder, the feeder name from variable feedernames will be included as
   %column one of this matrix. A for will be used from i=1->feedernames_size,
   %for filling this table with data. A nested for will rotate the reg_o
   %matrix looking for data from the feeder in the i row at the time and use
   %all its data to calculate SAIDI and SAIFI stuff.

   for i=1:feedernames_size(1,1)
      feederoutput(i+feederindex,3)=feedernames(i,1);

      for x=1:feederclients_size(1,1)

         if feederoutput(i+feederindex,3)==feederclients(x,1)
            feederoutput(i+feederindex,1)=r; %Copy R value (region raw data file index)
as region, this is used in case the feeder is non-existent in the feederclients database.
            feederoutput(i+feederindex,4)=feederclients(x,2); %Copy Clients
            feederoutput(i+feederindex,2)=feederclients(x,4); %Copy Right District
            feederoutput(i+feederindex,1)=feederclients(x,3);


            if feederoutput(i+feederindex,1)==feederclients(x,3)
               feederoutput(i+feederindex,2)=feederclients(x,4); %Copy Right District
               feederoutput(i+feederindex,4)=feederclients(x,2); %Copy Right Region


            end

         end
```

```
   end
   regdisoutput(i+feederindex,:)=feederoutput(i+feederindex,:);
end



for i=1:feedernames_size(1,1)
   a=1;
   b=1;
   c=1;
   d=1;
   e=1;
   f=1;
   g=1;
   MEDdatalog98=0;
   MEDdatalog99=0;
   MEDdatalog00=0;
   MEDdatalog01=0;
   MEDdatalog02=0;
   MEDdatalog03=0;
   MEDdatalog04=0;


   if i<feedernames_size(1,1)

      for j=feedernames(i,2):(feedernames(i+1,2)-1)

         % SAIDI----------use clients(row 12) and duration (row 7)

         %feederoutput(i,1)=feedernames(i,1);
         if reg_0(j,7)>momentary %If prevents counting reclosings as part of SAIDI
            %Division by #clients will be moved inside this if
            %because: 1. Sum of ratios == a ratio of sums
            %2. THis way i can divide by the normal number of total
            %clients or those specified in the database, which
            %could be higher due to use of interties.



            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
               if reg_0(j,12)>feederoutput(i+feederindex,4)
                  clients=reg_0(j,12);
               else
```

```
                clients=feederoutput(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end

            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutput(i+feederindex,2),reg_0(j,13),i
nputoption); %A %Adjustment step
            feederoutput(i+feederindex,8)=saididaily+feederoutput(i+feederindex,8);
%SAIDI
            saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutput(i+feederind
ex,2),reg_0(j,13),inputoption);

regdisoutput(i+feederindex,8)=saididailynumonly+regdisoutput(i+feederindex,8);

            if saididaily~=0

                MEDdatalog98(a,1)=log(saididaily);

                a=a+1;
            end
        end
        if reg_0(j,1)>=36161 && reg_0(j,1)<=36525
            if reg_0(j,12)>feederoutput(i+feederindex,4)
                clients=reg_0(j,12);
            else
                clients=feederoutput(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end

            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutput(i+feederindex,2),reg_0(j,13),i
nputoption); %Adjustment step
            feederoutput(i+feederindex,9)=saididaily+feederoutput(i+feederindex,9);
%SAIDI
            saididailynumonly=(reg_0(j,12)*reg_0(j,7));
```

```
saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutput(i+feederind
ex,2),reg_0(j,13),inputoption);


regdisoutput(i+feederindex,9)=saididailynumonly+regdisoutput(i+feederindex,9);

                    if saididaily~=0

                        MEDdatalog99(b,1)=log(saididaily);

                        b=b+1;
                    end
                end
                if reg_0(j,1)>=36526 && reg_0(j,1)<=36891
                    if reg_0(j,12)>feederoutput(i+feederindex,4)
                        clients=reg_0(j,12);
                    else
                        clients=feederoutput(i+feederindex,4);
                    end
                    if clients==0
                        clients=99;
                    end

                    saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutput(i+feederindex,2),reg_0(j,13),i
nputoption); %Adjustment step

feederoutput(i+feederindex,10)=saididaily+feederoutput(i+feederindex,10); %SAIDI
                    saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutput(i+feederind
ex,2),reg_0(j,13),inputoption);

regdisoutput(i+feederindex,10)=saididailynumonly+regdisoutput(i+feederindex,10);

                    if saididaily~=0

                        MEDdatalog00(c,1)=log(saididaily);

                        c=c+1;
                    end
                end
                if reg_0(j,1)>=36892 && reg_0(j,1)<=37256
```

```
                if reg_0(j,12)>feederoutput(i+feederindex,4)
                   clients=reg_0(j,12);
                else
                   clients=feederoutput(i+feederindex,4);
                end
                if clients==0
                   clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutput(i+feederindex,2),reg_0(j,13),i
nputoption); %Adjustment step

feederoutput(i+feederindex,11)=saididaily+feederoutput(i+feederindex,11); %SAIDI
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutput(i+feederind
ex,2),reg_0(j,13),inputoption);

regdisoutput(i+feederindex,11)=saididailynumonly+regdisoutput(i+feederindex,11);

                if saididaily~=0

                   MEDdatalog01(d,1)=log(saididaily);

                   d=d+1;
                end
             end
             if reg_0(j,1)>=37257 && reg_0(j,1)<=37621
                if reg_0(j,12)>feederoutput(i+feederindex,4)
                   clients=reg_0(j,12);
                else
                   clients=feederoutput(i+feederindex,4);
                end
                if clients==0
                   clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutput(i+feederindex,2),reg_0(j,13),i
nputoption); %Adjustment step

feederoutput(i+feederindex,12)=saididaily+feederoutput(i+feederindex,12); %SAIDI
```

```
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutput(i+feederind
ex,2),reg_0(j,13),inputoption);

regdisoutput(i+feederindex,12)=saididailynumonly+regdisoutput(i+feederindex,12);

                if saididaily~=0

                    MEDdatalog02(e,1)=log(saididaily);

                    e=e+1;
                end
            end
            if reg_0(j,1)>=37622 && reg_0(j,1)<=37986
                if reg_0(j,12)>feederoutput(i+feederindex,4)
                    clients=reg_0(j,12);
                else
                    clients=feederoutput(i+feederindex,4);
                end
                if clients==0
                    clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutput(i+feederindex,2),reg_0(j,13),i
nputoption); %Adjustment step

feederoutput(i+feederindex,13)=saididaily+feederoutput(i+feederindex,13); %SAIDI
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutput(i+feederind
ex,2),reg_0(j,13),inputoption);

regdisoutput(i+feederindex,13)=saididailynumonly+regdisoutput(i+feederindex,13);

                if saididaily~=0

                    MEDdatalog03(f,1)=log(saididaily);

                    f=f+1;
                end
            end
            if reg_0(j,1)>=37987 && reg_0(j,1)<=38352
```

```
                if reg_0(j,12)>feederoutput(i+feederindex,4)
                   clients=reg_0(j,12);
                else
                   clients=feederoutput(i+feederindex,4);
                end
                if clients==0
                   clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutput(i+feederindex,2),reg_0(j,13),i
nputoption); %Adjustment step

feederoutput(i+feederindex,14)=saididaily+feederoutput(i+feederindex,14); %SAIDI
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutput(i+feederind
ex,2),reg_0(j,13),inputoption);

regdisoutput(i+feederindex,14)=saididailynumonly+regdisoutput(i+feederindex,14);

                if saididaily~=0

                   MEDdatalog04(g,1)=log(saididaily);

                   g=g+1;
                end
             end




          end

       end
       %New MED module.....lets see if it works. :o
       %Need to chk feeder number for each iteration, search its
       %97MEDlog data, and copy it to the new log file.Thus, i need to
       %read from feederoutput, and read in a for feederMED.
       feederMED_size=size(feederMED);
       MEDdatalogA=0;
       t=1;
       for u=1:feederMED_size(1,1)
          if feederMED(u,1)==feederoutput(i+feederindex,3)
```

```
            MEDdatalogA(t,1)=feederMED(u,2);
            t=t+1;
        end
    end



    MEDdatalogB=vertcat(MEDdatalogA,MEDdatalog98);
    MEDdatalogC=vertcat(MEDdatalogB,MEDdatalog99);
    MEDdatalogD=vertcat(MEDdatalogC,MEDdatalog00);
    MEDdatalogE=vertcat(MEDdatalogD,MEDdatalog01);

MEDdatalogF=vertcat(MEDdatalog98,MEDdatalog99,MEDdatalog00,MEDdatalog01,M
EDdatalog02);

MEDdatalogG=vertcat(MEDdatalog99,MEDdatalog00,MEDdatalog01,MEDdatalog02,
MEDdatalog03);



    feederMEDtuki(i+feederindex,1)=feederoutput(i+feederindex,1);
    feederMEDtuki(i+feederindex,2)=feederoutput(i+feederindex,2);
    feederMEDtuki(i+feederindex,3)=feederoutput(i+feederindex,3);
    meantempA=mean(MEDdatalogA);
    devtempA=std(MEDdatalogA);
    combotempA=meantempA+2.5*devtempA;
    feederMEDtuki(i+feederindex,4)=exp(combotempA);

    meantempB=mean(MEDdatalogB);
    devtempB=std(MEDdatalogB);
    combotempB=meantempB+2.5*devtempB;
    feederMEDtuki(i+feederindex,5)=exp(combotempB);

    meantempC=mean(MEDdatalogC);
    devtempC=std(MEDdatalogC);
    combotempC=meantempC+2.5*devtempC;
    feederMEDtuki(i+feederindex,6)=exp(combotempC);

    meantempD=mean(MEDdatalogD);
    devtempD=std(MEDdatalogD);
    combotempD=meantempD+2.5*devtempD;
    feederMEDtuki(i+feederindex,7)=exp(combotempD);

    meantempE=mean(MEDdatalogE);
```

```
    devtempE=std(MEDdatalogE);
    combotempE=meantempE+2.5*devtempE;
    feederMEDtuki(i+feederindex,8)=exp(combotempE);

    meantempF=mean(MEDdatalogF);
    devtempF=std(MEDdatalogF);
    combotempF=meantempF+2.5*devtempF;
    feederMEDtuki(i+feederindex,9)=exp(combotempF);

    meantempG=mean(MEDdatalogG);
    devtempG=std(MEDdatalogG);
    combotempG=meantempG+2.5*devtempG;
    feederMEDtuki(i+feederindex,10)=exp(combotempG);

end
```

if i==feedernames_size(1,1) %If added to fix the problem when the last feeder was read on feedernames, which had no next feeder to use as end index in the previous IF.

```
    for j=feedernames(feedernames_size(1,1),2):reg_0_size(1,1)

        % SAIDI----------use clients(row 12) and duration (row 7)
        if reg_0(j,7)>momentary %If prevents counting reclosings as part of SAIDI

            %Division by #clients will be moved inside this if
            %because: 1. Sum of ratios == a ratio of sums
            %2. THis way i can divide by the normal number of total
            %clients or those specified in the database, which
            %could be higher due to use of interties.



            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
                if reg_0(j,12)>feederoutput(i+feederindex,4)
                    clients=reg_0(j,12);
                else
                    clients=feederoutput(i+feederindex,4);
                end
                if clients==0
                    clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
```

```
saididaily=saididailyWEIGHTER3(saididaily,feederoutput(i+feederindex,2),reg_0(j,13),i
nputoption); %Adjustment step
                feederoutput(i+feederindex,8)=saididaily+feederoutput(i+feederindex,8);
%SAIDI
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutput(i+feederind
ex,2),reg_0(j,13),inputoption);

regdisoutput(i+feederindex,8)=saididailynumonly+regdisoutput(i+feederindex,8);

                if saididaily~=0

                   MEDdatalog98(a,1)=log(saididaily);

                   a=a+1;
                end
             end
             if reg_0(j,1)>=36161 && reg_0(j,1)<=36525
                if reg_0(j,12)>feederoutput(i+feederindex,4)
                   clients=reg_0(j,12);
                else
                   clients=feederoutput(i+feederindex,4);
                end
                if clients==0
                   clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
                feederoutput(i+feederindex,9)=saididaily+feederoutput(i+feederindex,9);
%SAIDI
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutput(i+feederind
ex,2),reg_0(j,13),inputoption);

regdisoutput(i+feederindex,9)=saididailynumonly+regdisoutput(i+feederindex,9);

                if saididaily~=0

                   MEDdatalog99(b,1)=log(saididaily);

                   b=b+1;
                end
```

```
                end
                if reg_0(j,1)>=36526 && reg_0(j,1)<=36891
                    if reg_0(j,12)>feederoutput(i+feederindex,4)
                        clients=reg_0(j,12);
                    else
                        clients=feederoutput(i+feederindex,4);
                    end
                    if clients==0
                        clients=99;
                    end

                    saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutput(i+feederindex,2),reg_0(j,13),i
nputoption); %Adjustment step

feederoutput(i+feederindex,10)=saididaily+feederoutput(i+feederindex,10); %SAIDI
                    saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutput(i+feederind
ex,2),reg_0(j,13),inputoption);

regdisoutput(i+feederindex,10)=saididailynumonly+regdisoutput(i+feederindex,10);

                    if saididaily~=0

                        MEDdatalog00(c,1)=log(saididaily);

                        c=c+1;
                    end
                end
                if reg_0(j,1)>=36892 && reg_0(j,1)<=37256
                    if reg_0(j,12)>feederoutput(i+feederindex,4)
                        clients=reg_0(j,12);
                    else
                        clients=feederoutput(i+feederindex,4);
                    end
                    if clients==0
                        clients=99;
                    end

                    saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutput(i+feederindex,2),reg_0(j,13),i
nputoption); %Adjustment step
```

```
feederoutput(i+feederindex,11)=saididaily+feederoutput(i+feederindex,11); %SAIDI
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutput(i+feederind
ex,2),reg_0(j,13),inputoption);

regdisoutput(i+feederindex,11)=saididailynumonly+regdisoutput(i+feederindex,11);

                if saididaily~=0

                   MEDdatalog01(d,1)=log(saididaily);

                   d=d+1;
                end
             end
             if reg_0(j,1)>=37257 && reg_0(j,1)<=37621
                if reg_0(j,12)>feederoutput(i+feederindex,4)
                   clients=reg_0(j,12);
                else
                   clients=feederoutput(i+feederindex,4);
                end
                if clients==0
                   clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutput(i+feederindex,2),reg_0(j,13),i
nputoption); %Adjustment step

feederoutput(i+feederindex,12)=saididaily+feederoutput(i+feederindex,12); %SAIDI
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutput(i+feederind
ex,2),reg_0(j,13),inputoption);

regdisoutput(i+feederindex,12)=saididailynumonly+regdisoutput(i+feederindex,12);

                if saididaily~=0

                   MEDdatalog02(e,1)=log(saididaily);

                   e=e+1;
                end
```

```
            end
            if reg_0(j,1)>=37622 && reg_0(j,1)<=37986
               if reg_0(j,12)>feederoutput(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutput(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end

               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutput(i+feederindex,2),reg_0(j,13),i
nputoption); %Adjustment step

feederoutput(i+feederindex,13)=saididaily+feederoutput(i+feederindex,13); %SAIDI
               saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutput(i+feederind
ex,2),reg_0(j,13),inputoption);

regdisoutput(i+feederindex,13)=saididailynumonly+regdisoutput(i+feederindex,13);

               if saididaily~=0

                  MEDdatalog03(f,1)=log(saididaily);

                  f=f+1;
               end
            end
            if reg_0(j,1)>=37987 && reg_0(j,1)<=38352
               if reg_0(j,12)>feederoutput(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutput(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end

               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutput(i+feederindex,2),reg_0(j,13),i
nputoption); %Adjustment step
```

```
feederoutput(i+feederindex,14)=saididaily+feederoutput(i+feederindex,14); %SAIDI
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutput(i+feederind
ex,2),reg_0(j,13),inputoption);

regdisoutput(i+feederindex,14)=saididailynumonly+regdisoutput(i+feederindex,14);

                if saididaily~=0

                   MEDdatalog04(g,1)=log(saididaily);

                   g=g+1;
                end
             end
          end

       end
       MEDdatalogA=0;
       t=1;
       for u=1:feederMED_size(1,1)
          if feederMED(u,1)==feederoutput(i+feederindex,3)
             MEDdatalogA(t,1)=feederMED(u,2);
             t=t+1;
          end
       end


       MEDdatalogB=vertcat(MEDdatalogA,MEDdatalog98);
       MEDdatalogC=vertcat(MEDdatalogB,MEDdatalog99);
       MEDdatalogD=vertcat(MEDdatalogC,MEDdatalog00);
       MEDdatalogE=vertcat(MEDdatalogD,MEDdatalog01);

MEDdatalogF=vertcat(MEDdatalog98,MEDdatalog99,MEDdatalog00,MEDdatalog01,M
EDdatalog02);

MEDdatalogG=vertcat(MEDdatalog99,MEDdatalog00,MEDdatalog01,MEDdatalog02,
MEDdatalog03);

       feederMEDtuki(i+feederindex,1)=feederoutput(i+feederindex,1);
       feederMEDtuki(i+feederindex,2)=feederoutput(i+feederindex,2);
       feederMEDtuki(i+feederindex,3)=feederoutput(i+feederindex,3);
       meantempA=mean(MEDdatalogA);
       devtempA=std(MEDdatalogA);
```

```
    combotempA=meantempA+2.5*devtempA;
    feederMEDtuki(i+feederindex,4)=exp(combotempA);

    meantempB=mean(MEDdatalogB);
    devtempB=std(MEDdatalogB);
    combotempB=meantempB+2.5*devtempB;
    feederMEDtuki(i+feederindex,5)=exp(combotempB);

    meantempC=mean(MEDdatalogC);
    devtempC=std(MEDdatalogC);
    combotempC=meantempC+2.5*devtempC;
    feederMEDtuki(i+feederindex,6)=exp(combotempC);

    meantempD=mean(MEDdatalogD);
    devtempD=std(MEDdatalogD);
    combotempD=meantempD+2.5*devtempD;
    feederMEDtuki(i+feederindex,7)=exp(combotempD);

    meantempE=mean(MEDdatalogE);
    devtempE=std(MEDdatalogE);
    combotempE=meantempE+2.5*devtempE;
    feederMEDtuki(i+feederindex,8)=exp(combotempE);

    meantempF=mean(MEDdatalogF);
    devtempF=std(MEDdatalogF);
    combotempF=meantempF+2.5*devtempF;
    feederMEDtuki(i+feederindex,9)=exp(combotempF);

    meantempG=mean(MEDdatalogG);
    devtempG=std(MEDdatalogG);
    combotempG=meantempG+2.5*devtempG;
    feederMEDtuki(i+feederindex,10)=exp(combotempG);



end

if feederoutput(i+feederindex,4)==0
    feederoutput(i+feederindex,8)=999;
    feederoutput(i+feederindex,9)=999;
    feederoutput(i+feederindex,10)=999;
    feederoutput(i+feederindex,11)=999;
    feederoutput(i+feederindex,12)=999;
    feederoutput(i+feederindex,13)=999;
    feederoutput(i+feederindex,14)=999;
```

```
        end
    end


    feederoutput_size=size(feederoutput);

    feederindex=feederindex+feedernames_size(1,1);

end
%-------------------------------------------------------------------------%
%////////////////////////////END OF FEEDEROUPUT MED/////////////////--3-%
%-------------------------------------------------------------------------%


%-------------------------------------------------------------------------%
%////////////////////////////ALL MEDDED SAIDI SAIFI/////////////////--4-%
%-------------------------------------------------------------------------%
feederoutputMEDDED=zeros(feederoutput_size);
feederindex=0;
for r=0:6
    if r==0
        reg_0=region_0;
        regionmarker(1,1)=1;
        saididailytemp=reg0_saididaily;
        saididailytempMEDDED=reg0_saididailyMEDDED;
        r
    end
    if r==1
        reg_0=region_1;
        regionmarker(2,1)=feederindex;
        saididailytemp=reg1_saididaily;
        saididailytempMEDDED=reg1_saididailyMEDDED;
        r
    end
    if r==2
        reg_0=region_2;
        regionmarker(3,1)=feederindex;
        saididailytemp=reg2_saididaily;
        saididailytempMEDDED=reg2_saididailyMEDDED;
        r
    end
    if r==3
        reg_0=region_3;
        regionmarker(4,1)=feederindex;
        saididailytemp=reg3_saididaily;
```

```
   saididailytempMEDDED=reg3_saididailyMEDDED;
   r
end
if r==4
   reg_0=region_4;
   regionmarker(5,1)=feederindex;
   saididailytemp=reg4_saididaily;
   saididailytempMEDDED=reg4_saididailyMEDDED;
   r
end
if r==5
   reg_0=region_5;
   regionmarker(6,1)=feederindex;
   saididailytemp=reg5_saididaily;
   saididailytempMEDDED=reg5_saididailyMEDDED;
   r
end
if r==6
   reg_0=region_6;
   regionmarker(7,1)=feederindex;
   saididailytemp=reg6_saididaily;
   saididailytempMEDDED=reg6_saididailyMEDDED;
   r
end

reg_0_size=size(reg_0);
feedernames=zeros(2,3);
feedernames_size=size(feedernames);

%---This for will read all the feeder column and extract the individual
%feeder names, which will be later used for sorting.

for i=1:reg_0_size(1,1)

   if i==1
      j=1;
      feedernames(j,1)=reg_0(i,11);
      feedernames(j,2)=1;
      feedernames(j,3)=reg_0(i,10);


   end
   if i>1 && reg_0(i,11)~=reg_0(i-1,11)
      j=j+1;
```

```
        feedernames(j,1)=reg_0(i,11);
        feedernames(j,2)=i; %Stores FIRST row where this feeder appears in XLS file.
        feedernames(j,3)=reg_0(i,10);

      end



   end

   feedernames_size=size(feedernames);
   %Create a matrix to hold the feeder data (feederoutput). One row will be created for each
   %feeder, the feeder name from variable feedernames will be included as
   %column one of this matrix. A for will be used from i=1->feedernames_size,
   %for filling this table with data. A nested for will rotate the reg_o
   %matrix looking for data from the feeder in the i row at the time and use
   %all its data to calculate SAIDI and SAIFI stuff.

   for i=1:feedernames_size(1,1)
      feederoutputMEDDED(i+feederindex,3)=feedernames(i,1);
      %feederoutputMEDDED(i+feederindex,2)=feedernames(i,3);

      for x=1:feederclients_size(1,1)

         if feederoutputMEDDED(i+feederindex,3)==feederclients(x,1)
            feederoutputMEDDED(i+feederindex,1)=r; %Copy R value (region raw data file index) as region, this is used in case the feeder is non-existent in the feederclients database.
            feederoutputMEDDED(i+feederindex,4)=feederclients(x,2); %Copy Clients
            feederoutputMEDDED(i+feederindex,2)=feederclients(x,4); %Copy Right District
            feederoutputMEDDED(i+feederindex,1)=feederclients(x,3);


            if feederoutputMEDDED(i+feederindex,1)==feederclients(x,3)
               feederoutputMEDDED(i+feederindex,2)=feederclients(x,4); %Copy Right District
               feederoutputMEDDED(i+feederindex,4)=feederclients(x,2); %Copy Right Region


            end

         end
```

```
    end
    regdisoutputMEDDED(i+feederindex,:)=feederoutputMEDDED(i+feederindex,:);
  end


%*********************************************************************
***
  %-----SAIDI W/O MED + FAULT CODE EXTRACTION-----------------

%*********************************************************************
***


  for i=1:feedernames_size(1,1)
    feederFAULT_append(i+feederindex,1)=feederoutput(i+feederindex,1);
    feederFAULT_append(i+feederindex,2)=feederoutput(i+feederindex,2);
    feederFAULT_append(i+feederindex,3)=feederoutput(i+feederindex,3);

    if i<feedernames_size(1,1)

      for j=feedernames(i,2):(feedernames(i+1,2)-1)

        % SAIDI----------use clients(row 12) and duration (row 7)

        %feederoutputMEDDED(i,1)=feedernames(i,1);
        if reg_0(j,7)>momentary %If prevents counting reclosings as part of SAIDI
          %Division by #clients will be moved inside this if
          %because: 1. Sum of ratios == a ratio of sums
          %2. THis way i can divide by the normal number of total
          %clients or those specified in the database, which
          %could be higher due to use of interties.


          if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
            if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
              clients=reg_0(j,12);
            else
              clients=feederoutputMEDDED(i+feederindex,4);
            end
            if clients==0
              clients=99;
            end
            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
```

```
saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
                if clients>0
                   saididailytemp(j,1)=reg_0(j,11);
                   saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytemp(j,3)=clients;
                   saididailytemp(j,4)=saididailynumonly;
                end
                if (saididaily>=feederMEDtuki(i+feederindex,4))&&(r~=4)
                   saididaily=0;
                   saididailynumonly=0;
                   MEDcount98=MEDcount98+1;
                end
                if clients<=50
                   saididaily=0;
                   saididailynumonly=0;
                end
                if reg_0(j,13)==38 || reg_0(j,13)==39 %Elimina operaciones en el 38 o 115
                   saididaily=0;
                   saididailynumonly=0;

                end

feederoutputMEDDED(i+feederindex,8)=saididaily+feederoutputMEDDED(i+feederindex,8); %SAIDI

regdisoutputMEDDED(i+feederindex,8)=saididailynumonly+regdisoutputMEDDED(i+feederindex,8);
                if clients>0
                   saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytempMEDDED(j,3)=clients;
                   saididailytempMEDDED(j,4)=saididailynumonly;
                end


             end
             if reg_0(j,1)>=36161 && reg_0(j,1)<=36525
```

```
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                   clients=reg_0(j,12);
                else
                   clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                   clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
                if clients>0
                   saididailytemp(j,1)=reg_0(j,11);
                   saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytemp(j,3)=clients;
                   saididailytemp(j,5)=saididailynumonly;
                end
                if saididaily>=feederMEDtuki(i+feederindex,5)
                   saididaily=0;
                   saididailynumonly=0;
                   MEDcount99=MEDcount99+1;
                end
                if clients<=50
                   saididaily=0;
                   saididailynumonly=0;
                end
                if reg_0(j,13)==38 || reg_0(j,13)==39
                   saididaily=0;
                   saididailynumonly=0;
                end

feederoutputMEDDED(i+feederindex,9)=saididaily+feederoutputMEDDED(i+feederindex,9); %SAIDI

regdisoutputMEDDED(i+feederindex,9)=saididailynumonly+regdisoutputMEDDED(i+feederindex,9);
                if clients>0
                   saididailytempMEDDED(j,1)=reg_0(j,11);
```

```
saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                    saididailytempMEDDED(j,3)=clients;
                    saididailytempMEDDED(j,5)=saididailynumonly;
                end



            end
            if reg_0(j,1)>=36526 && reg_0(j,1)<=36891
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                    clients=reg_0(j,12);
                else
                    clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                    clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutputMEDDED(i
+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
                if clients>0
                    saididailytemp(j,1)=reg_0(j,11);
                    saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                    saididailytemp(j,3)=clients;
                    saididailytemp(j,6)=saididailynumonly;
                end
                if saididaily>=feederMEDtuki(i+feederindex,6)
                    saididaily=0;
                    saididailynumonly=0;
                    MEDcount00=MEDcount00+1;
                end
                if clients<=50
                    saididaily=0;
                    saididailynumonly=0;
                end
                if reg_0(j,13)==38 || reg_0(j,13)==39
                    saididaily=0;
                    saididailynumonly=0;
```

```
                end

feederoutputMEDDED(i+feederindex,10)=saididaily+feederoutputMEDDED(i+feederin
dex,10); %SAIDI

regdisoutputMEDDED(i+feederindex,10)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,10);
                if clients>0
                   saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytempMEDDED(j,3)=clients;
                   saididailytempMEDDED(j,6)=saididailynumonly;
                end


             end
             if reg_0(j,1)>=36892 && reg_0(j,1)<=37256
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                   clients=reg_0(j,12);
                else
                   clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                   clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutputMEDDED(i
+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
                if clients>0
                   saididailytemp(j,1)=reg_0(j,11);
                   saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytemp(j,3)=clients;
                   saididailytemp(j,7)=saididailynumonly;
                end
                if saididaily>=feederMEDtuki(i+feederindex,7)
                   saididaily=0;
                   saididailynumonly=0;
```

```
                MEDcount01=MEDcount01+1;
            end
            if clients<=50
                saididaily=0;
                saididailynumonly=0;
            end
            if reg_0(j,13)==38 || reg_0(j,13)==39
                saididaily=0;
                saididailynumonly=0;

            end

feederoutputMEDDED(i+feederindex,11)=saididaily+feederoutputMEDDED(i+feederin
dex,11); %SAIDI

regdisoutputMEDDED(i+feederindex,11)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,11);
            if clients>0
                saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytempMEDDED(j,3)=clients;
                saididailytempMEDDED(j,7)=saididailynumonly;
            end


        end
        if reg_0(j,1)>=37257 && reg_0(j,1)<=37621
            if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                clients=reg_0(j,12);
            else
                clients=feederoutputMEDDED(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end

            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step
            saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutputMEDDED(i
+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
```

```
                if clients>0
                   saididailytemp(j,1)=reg_0(j,11);
                   saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytemp(j,3)=clients;
                   saididailytemp(j,8)=saididailynumonly;
                end
                if saididaily>=feederMEDtuki(i+feederindex,8)
                   saididaily=0;
                   saididailynumonly=0;
                   MEDcount02=MEDcount02+1;
                end
                if clients<=50
                   saididaily=0;
                   saididailynumonly=0;
                end
                if reg_0(j,13)==38 || reg_0(j,13)==39
                   saididaily=0;
                   saididailynumonly=0;

                end

feederoutputMEDDED(i+feederindex,12)=saididaily+feederoutputMEDDED(i+feederin
dex,12); %SAIDI

regdisoutputMEDDED(i+feederindex,12)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,12);
                if clients>0
                   saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytempMEDDED(j,3)=clients;
                   saididailytempMEDDED(j,8)=saididailynumonly;
                end

             end
             if reg_0(j,1)>=37622 && reg_0(j,1)<=37986
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                   clients=reg_0(j,12);
                else
                   clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                   clients=99;
                end
```

```
                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutputMEDDED(i
+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
                if clients>0
                   saididailytemp(j,1)=reg_0(j,11);
                   saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytemp(j,3)=clients;
                   saididailytemp(j,9)=saididailynumonly;
                end
                if saididaily>=feederMEDtuki(i+feederindex,9)
                   saididaily=0;
                   saididailynumonly=0;
                   MEDcount03=MEDcount03+1;
                end
                if clients<=50
                   saididaily=0;
                   saididailynumonly=0;
                end
                if reg_0(j,13)==38 || reg_0(j,13)==39
                   saididaily=0;
                   saididailynumonly=0;

                end

feederoutputMEDDED(i+feederindex,13)=saididaily+feederoutputMEDDED(i+feederin
dex,13); %SAIDI

regdisoutputMEDDED(i+feederindex,13)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,13);
                if clients>0
                   saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytempMEDDED(j,3)=clients;
                   saididailytempMEDDED(j,9)=saididailynumonly;
                end



             end
```

```
            if reg_0(j,1)>=37987 && reg_0(j,1)<=38352
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end

               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step
               saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutputMEDDED(i
+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
               if clients>0
                  saididailytemp(j,1)=reg_0(j,11);
                  saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                  saididailytemp(j,3)=clients;
                  saididailytemp(j,10)=saididailynumonly;
               end
               if saididaily>=feederMEDtuki(i+feederindex,10)
                  saididaily=0;
                  saididailynumonly=0;
                  MEDcount04=MEDcount04+1;
               end
               if clients<=50
                  saididaily=0;
                  saididailynumonly=0;
               end
               if reg_0(j,13)==38 || reg_0(j,13)==39
                  saididaily=0;
                  saididailynumonly=0;

               end

feederoutputMEDDED(i+feederindex,14)=saididaily+feederoutputMEDDED(i+feederin
dex,14); %SAIDI

regdisoutputMEDDED(i+feederindex,14)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,14);
               if clients>0
```

```
                saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytempMEDDED(j,3)=clients;
                saididailytempMEDDED(j,10)=saididailynumonly;
              end

            end

          end

        end
```

if i==feedernames_size(1,1) %If added to fix the problem when the last feeder was read on feedernames, which had no next feeder to use as end index in the previous IF.

```
        for j=feedernames(feedernames_size(1,1),2):reg_0_size(1,1)


          % SAIDI----------use clients(row 12) and duration (row 7)
          if reg_0(j,7)>momentary %If prevents counting reclosings as part of SAIDI

            %Division by #clients will be moved inside this if
            %because: 1. Sum of ratios == a ratio of sums
            %2. THis way i can divide by the normal number of total
            %clients or those specified in the database, which
            %could be higher due to use of interties.



            if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
              if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                clients=reg_0(j,12);
              else
                clients=feederoutputMEDDED(i+feederindex,4);
              end
              if clients==0
                clients=99;
              end

              saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
```

```
saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
                if clients>0
                   saididailytemp(j,1)=reg_0(j,11);
                   saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytemp(j,3)=clients;
                   saididailytemp(j,4)=saididailynumonly;
                end
                if (saididaily>=feederMEDtuki(i+feederindex,4))&&(r~=4)
                   saididaily=0;
                   saididailynumonly=0;
                   MEDcount98=MEDcount98+1;
                end
                if clients<=50
                   saididaily=0;
                   saididailynumonly=0;
                end
                if reg_0(j,13)==38 || reg_0(j,13)==39
                   saididaily=0;
                   saididailynumonly=0;

                end

feederoutputMEDDED(i+feederindex,8)=saididaily+feederoutputMEDDED(i+feederindex,8); %SAIDI

regdisoutputMEDDED(i+feederindex,8)=saididailynumonly+regdisoutputMEDDED(i+feederindex,8);
                if clients>0
                   saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytempMEDDED(j,3)=clients;
                   saididailytempMEDDED(j,4)=saididailynumonly;
                end



            end
            if reg_0(j,1)>=36161 && reg_0(j,1)<=36525
```

```
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,5)
                   clients=reg_0(j,12);
                else
                   clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                   clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutputMEDDED(i
+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
                if clients>0
                   saididailytemp(j,1)=reg_0(j,11);
                   saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytemp(j,3)=clients;
                   saididailytemp(j,5)=saididailynumonly;
                end
                if saididaily>=feederMEDtuki(i+feederindex,6)
                   saididaily=0;
                   saididailynumonly=0;
                   MEDcount99=MEDcount99+1;
                end
                if clients<=50
                   saididaily=0;
                   saididailynumonly=0;
                end
                if reg_0(j,13)==38 || reg_0(j,13)==39
                   saididaily=0;
                   saididailynumonly=0;

                end

feederoutputMEDDED(i+feederindex,9)=saididaily+feederoutputMEDDED(i+feederind
ex,9); %SAIDI

regdisoutputMEDDED(i+feederindex,9)=saididailynumonly+regdisoutputMEDDED(i+f
eederindex,9);
                if clients>0
                   saididailytempMEDDED(j,1)=reg_0(j,11);
```

```
saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                    saididailytempMEDDED(j,3)=clients;
                    saididailytempMEDDED(j,5)=saididailynumonly;
                end



            end
            if reg_0(j,1)>=36526 && reg_0(j,1)<=36891
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                    clients=reg_0(j,12);
                else
                    clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                    clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutputMEDDED(i
+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
                if clients>0
                    saididailytemp(j,1)=reg_0(j,11);
                    saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                    saididailytemp(j,3)=clients;
                    saididailytemp(j,6)=saididailynumonly;
                end
                if saididaily>=feederMEDtuki(i+feederindex,7)
                    saididaily=0;
                    saididailynumonly=0;
                    MEDcount00=MEDcount00+1;
                end
                if clients<=50
                    saididaily=0;
                    saididailynumonly=0;
                end
                if reg_0(j,13)==38 || reg_0(j,13)==39
                    saididaily=0;
                    saididailynumonly=0;
```

```
                end

feederoutputMEDDED(i+feederindex,10)=saididaily+feederoutputMEDDED(i+feederin
dex,10); %SAIDI

regdisoutputMEDDED(i+feederindex,10)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,10);
                if clients>0
                   saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytempMEDDED(j,3)=clients;
                   saididailytempMEDDED(j,6)=saididailynumonly;
                end



             end
             if reg_0(j,1)>=36892 && reg_0(j,1)<=37256
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                   clients=reg_0(j,12);
                else
                   clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                   clients=99;
                end

                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step
                saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutputMEDDED(i
+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
                if clients>0
                   saididailytemp(j,1)=reg_0(j,11);
                   saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytemp(j,3)=clients;
                   saididailytemp(j,7)=saididailynumonly;
                end
                if saididaily>=feederMEDtuki(i+feederindex,8)
                   saididaily=0;
```

```
                saididailynumonly=0;
                MEDcount01=MEDcount01+1;
            end
            if clients<=50
                saididaily=0;
                saididailynumonly=0;
            end
            if reg_0(j,13)==38 || reg_0(j,13)==39
                saididaily=0;
                saididailynumonly=0;

            end

feederoutputMEDDED(i+feederindex,11)=saididaily+feederoutputMEDDED(i+feederin
dex,11); %SAIDI

regdisoutputMEDDED(i+feederindex,11)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,11);
            if clients>0
                saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytempMEDDED(j,3)=clients;
                saididailytempMEDDED(j,7)=saididailynumonly;
            end




        end
        if reg_0(j,1)>=37257 && reg_0(j,1)<=37621
            if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                clients=reg_0(j,12);
            else
                clients=feederoutputMEDDED(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end

            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step
            saididailynumonly=(reg_0(j,12)*reg_0(j,7));
```

```
saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutputMEDDED(i
+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
                if clients>0
                   saididailytemp(j,1)=reg_0(j,11);
                   saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytemp(j,3)=clients;
                   saididailytemp(j,8)=saididailynumonly;
                end
                if saididaily>=feederMEDtuki(i+feederindex,9)
                   saididaily=0;
                   saididailynumonly=0;
                   MEDcount02=MEDcount02+1;
                end
                if clients<=50
                   saididaily=0;
                   saididailynumonly=0;
                end
                if reg_0(j,13)==38 || reg_0(j,13)==39
                   saididaily=0;
                   saididailynumonly=0;

                end

feederoutputMEDDED(i+feederindex,12)=saididaily+feederoutputMEDDED(i+feederin
dex,12); %SAIDI

regdisoutputMEDDED(i+feederindex,12)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,12);
                if clients>0
                   saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                   saididailytempMEDDED(j,3)=clients;
                   saididailytempMEDDED(j,8)=saididailynumonly;
                end




             end
             if reg_0(j,1)>=37622 && reg_0(j,1)<=37986
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                   clients=reg_0(j,12);
                else
```

```
                clients=feederoutputMEDDED(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end

            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step
            saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutputMEDDED(i
+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
            if clients>0
                saididailytemp(j,1)=reg_0(j,11);
                saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytemp(j,3)=clients;
                saididailytemp(j,9)=saididailynumonly;
            end
            if saididaily>=feederMEDtuki(i+feederindex,10)
                saididaily=0;
                saididailynumonly=0;
                MEDcount03=MEDcount03+1;
            end
            if clients<=50
                saididaily=0;
                saididailynumonly=0;
            end
            if reg_0(j,13)==38 || reg_0(j,13)==39
                saididaily=0;
                saididailynumonly=0;

            end

feederoutputMEDDED(i+feederindex,13)=saididaily+feederoutputMEDDED(i+feederin
dex,13); %SAIDI

regdisoutputMEDDED(i+feederindex,13)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,13);
            if clients>0
                saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytempMEDDED(j,3)=clients;
```

```
                saididailytempMEDDED(j,9)=saididailynumonly;
            end




        end
        if reg_0(j,1)>=37987 && reg_0(j,1)<=38352
            if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                clients=reg_0(j,12);
            else
                clients=feederoutputMEDDED(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end

            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step
            saididailynumonly=(reg_0(j,12)*reg_0(j,7));

saididailynumonly=saididailyWEIGHTER3(saididailynumonly,feederoutputMEDDED(i
+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
            if clients>0
                saididailytemp(j,1)=reg_0(j,11);
                saididailytemp(j,2)=feederoutputMEDDED(i+feederindex,2);
                saididailytemp(j,3)=clients;
                saididailytemp(j,10)=saididailynumonly;
            end
            if saididaily>=feederMEDtuki(i+feederindex,9)
                saididaily=0;
                saididailynumonly=0;
                MEDcount04=MEDcount04+1;
            end
            if clients<=50
                saididaily=0;
                saididailynumonly=0;
            end
            if reg_0(j,13)==38 || reg_0(j,13)==39
                saididaily=0;
                saididailynumonly=0;

            end
```

```
feederoutputMEDDED(i+feederindex,14)=saididaily+feederoutputMEDDED(i+feederin
dex,14); %SAIDI

regdisoutputMEDDED(i+feederindex,14)=saididailynumonly+regdisoutputMEDDED(i+
feederindex,14);
                    if clients>0
                       saididailytempMEDDED(j,1)=reg_0(j,11);

saididailytempMEDDED(j,2)=feederoutputMEDDED(i+feederindex,2);
                       saididailytempMEDDED(j,3)=clients;
                       saididailytempMEDDED(j,10)=saididailynumonly;
                    end




                 end

              end

           end

        end

        if feederoutputMEDDED(i+feederindex,4)==0
           feederoutputMEDDED(i+feederindex,8)=999;
           feederoutputMEDDED(i+feederindex,9)=999;
           feederoutputMEDDED(i+feederindex,10)=999;
           feederoutputMEDDED(i+feederindex,11)=999;
           feederoutputMEDDED(i+feederindex,12)=999;
           feederoutputMEDDED(i+feederindex,13)=999;
           feederoutputMEDDED(i+feederindex,14)=999;
        end




     end
```

```
%*********************************************************************
****
   %----SAIFI W/O MED CALCULATION-----------------------------------------

%*********************************************************************
****

   for i=1:feedernames_size(1,1)

      if i<feedernames_size(1,1)
         for j=feedernames(i,2):(feedernames(i+1,2)-1)

            % SAIFI----------use affected clients(row 12 reg_0) and total clients (row 3
feederclients)
            %Saifi will be calculated with first instance of an interruption. If
            %sectionalization occurs, this subsequent smaller interruptions will not be
            %used for calculation purposes.
            if (j==1)&&(reg_0(j,7)>momentary)

               if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
                  if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                     clients=reg_0(j,12);
                  else
                     clients=feederoutputMEDDED(i+feederindex,4);
                  end
                  if clients==0
                     clients=99;
                  end
                  saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step

                  if (saididaily<feederMEDtuki(i+feederindex,4))&&(r~=4)&&(clients>50)

feederoutputMEDDED(i+feederindex,16)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,16);

regdisoutputMEDDED(i+feederindex,16)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,16);
                  end

feederoutput(i+feederindex,16)=(reg_0(j,12))/clients+feederoutput(i+feederindex,16);
```

```
regdisoutput(i+feederindex,16)=(reg_0(j,12))+regdisoutput(i+feederindex,16);


                end
                if reg_0(j,1)>=36161 && reg_0(j,1)<=36525
                   if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                      clients=reg_0(j,12);
                   else
                      clients=feederoutputMEDDED(i+feederindex,4);
                   end
                   if clients==0
                      clients=99;
                   end
                   saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step

                   if saididaily<feederMEDtuki(i+feederindex,5)&&(clients>50)

feederoutputMEDDED(i+feederindex,17)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,17);

regdisoutputMEDDED(i+feederindex,17)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,17);
                   end

feederoutput(i+feederindex,17)=(reg_0(j,12))/clients+feederoutput(i+feederindex,17);

regdisoutput(i+feederindex,17)=(reg_0(j,12))+regdisoutput(i+feederindex,17);

                end
                if reg_0(j,1)>=36526 && reg_0(j,1)<=36891
                   if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                      clients=reg_0(j,12);
                   else
                      clients=feederoutputMEDDED(i+feederindex,4);
                   end
                   if clients==0
                      clients=99;
                   end
                   saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
```

```
saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step

                if saididaily<feederMEDtuki(i+feederindex,6)&&(clients>50)

feederoutputMEDDED(i+feederindex,18)=(reg_0(j,12))/clients+feederoutputMEDDED(i+feederindex,18);

regdisoutputMEDDED(i+feederindex,18)=(reg_0(j,12))+regdisoutputMEDDED(i+feederindex,18);
                end

feederoutput(i+feederindex,18)=(reg_0(j,12))/clients+feederoutput(i+feederindex,18);

regdisoutput(i+feederindex,18)=(reg_0(j,12))+regdisoutput(i+feederindex,18);

            end
            if reg_0(j,1)>=36892 && reg_0(j,1)<=37256
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end
               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step

                if saididaily<feederMEDtuki(i+feederindex,7)&&(clients>50)

feederoutputMEDDED(i+feederindex,19)=(reg_0(j,12))/clients+feederoutputMEDDED(i+feederindex,19);

regdisoutputMEDDED(i+feederindex,19)=(reg_0(j,12))+regdisoutputMEDDED(i+feederindex,19);
                end

feederoutput(i+feederindex,19)=(reg_0(j,12))/clients+feederoutput(i+feederindex,19);

regdisoutput(i+feederindex,19)=(reg_0(j,12))+regdisoutput(i+feederindex,19);
```

```
            end
            if reg_0(j,1)>=37257 && reg_0(j,1)<=37621
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end
               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step

               if saididaily<feederMEDtuki(i+feederindex,8)&&(clients>50)

feederoutputMEDDED(i+feederindex,20)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,20);

regdisoutputMEDDED(i+feederindex,20)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,20);
               end

feederoutput(i+feederindex,20)=(reg_0(j,12))/clients+feederoutput(i+feederindex,20);

regdisoutput(i+feederindex,20)=(reg_0(j,12))+regdisoutput(i+feederindex,20);

            end
            if reg_0(j,1)>=37622 && reg_0(j,1)<=37986
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end
               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step

               if saididaily<feederMEDtuki(i+feederindex,9)&&(clients>50)
```

```
feederoutputMEDDED(i+feederindex,21)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,21);

regdisoutputMEDDED(i+feederindex,21)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,21);
                end

feederoutput(i+feederindex,21)=(reg_0(j,12))/clients+feederoutput(i+feederindex,21);

regdisoutput(i+feederindex,21)=(reg_0(j,12))+regdisoutput(i+feederindex,21);

            end
            if reg_0(j,1)>=37987 && reg_0(j,1)<=38352
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end
               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step

               if saididaily<feederMEDtuki(i+feederindex,10)&&(clients>50)

feederoutputMEDDED(i+feederindex,22)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,22);

regdisoutputMEDDED(i+feederindex,22)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,22);
               end

feederoutput(i+feederindex,22)=(reg_0(j,12))/clients+feederoutput(i+feederindex,22);

regdisoutput(i+feederindex,22)=(reg_0(j,12))+regdisoutput(i+feederindex,22);

            end
         end

         if j>1
            if ((reg_0(j,7)>momentary)&&(reg_0(j,2)~=reg_0(j-1,2)))
```

```
                if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
                   if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                      clients=reg_0(j,12);
                   else
                      clients=feederoutputMEDDED(i+feederindex,4);
                   end
                   if clients==0
                      clients=99;
                   end
                   saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step

                   if
(saididaily<feederMEDtuki(i+feederindex,4))&&(r~=4)&&(clients>50)

feederoutputMEDDED(i+feederindex,16)=(reg_0(j,12))/clients+feederoutputMEDDED(i+feederindex,16);

regdisoutputMEDDED(i+feederindex,16)=(reg_0(j,12))+regdisoutputMEDDED(i+feederindex,16);
                   end

feederoutput(i+feederindex,16)=(reg_0(j,12))/clients+feederoutput(i+feederindex,16);

regdisoutput(i+feederindex,16)=(reg_0(j,12))+regdisoutput(i+feederindex,16);

                end
                if reg_0(j,1)>=36161 && reg_0(j,1)<=36525
                   if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                      clients=reg_0(j,12);
                   else
                      clients=feederoutputMEDDED(i+feederindex,4);
                   end
                   if clients==0
                      clients=99;
                   end
                   saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step

                   if saididaily<feederMEDtuki(i+feederindex,5)&&(clients>50)
```

```
feederoutputMEDDED(i+feederindex,17)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,17);

regdisoutputMEDDED(i+feederindex,17)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,17);
                    end

feederoutput(i+feederindex,17)=(reg_0(j,12))/clients+feederoutput(i+feederindex,17);

regdisoutput(i+feederindex,17)=(reg_0(j,12))+regdisoutput(i+feederindex,17);

                end
                if reg_0(j,1)>=36526 && reg_0(j,1)<=36891
                   if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                      clients=reg_0(j,12);
                   else
                      clients=feederoutputMEDDED(i+feederindex,4);
                   end
                   if clients==0
                      clients=99;
                   end
                   saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step

                    if saididaily<feederMEDtuki(i+feederindex,6)&&(clients>50)

feederoutputMEDDED(i+feederindex,18)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,18);

regdisoutputMEDDED(i+feederindex,18)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,18);
                    end

feederoutput(i+feederindex,18)=(reg_0(j,12))/clients+feederoutput(i+feederindex,18);

regdisoutput(i+feederindex,18)=(reg_0(j,12))+regdisoutput(i+feederindex,18);

                end
                if reg_0(j,1)>=36892 && reg_0(j,1)<=37256
                   if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                      clients=reg_0(j,12);
                   else
```

```
                clients=feederoutputMEDDED(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end
            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step

            if saididaily<feederMEDtuki(i+feederindex,7)&&(clients>50)

feederoutputMEDDED(i+feederindex,19)=(reg_0(j,12))/clients+feederoutputMEDDED(i+feederindex,19);

regdisoutputMEDDED(i+feederindex,19)=(reg_0(j,12))+regdisoutputMEDDED(i+feederindex,19);
            end

feederoutput(i+feederindex,19)=(reg_0(j,12))/clients+feederoutput(i+feederindex,19);

regdisoutput(i+feederindex,19)=(reg_0(j,12))+regdisoutput(i+feederindex,19);

        end
        if reg_0(j,1)>=37257 && reg_0(j,1)<=37621
            if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                clients=reg_0(j,12);
            else
                clients=feederoutputMEDDED(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end
            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step

            if saididaily<feederMEDtuki(i+feederindex,8)&&(clients>50)

feederoutputMEDDED(i+feederindex,20)=(reg_0(j,12))/clients+feederoutputMEDDED(i+feederindex,20);

regdisoutputMEDDED(i+feederindex,20)=(reg_0(j,12))+regdisoutputMEDDED(i+feederindex,20);
```

```
                end

feederoutput(i+feederindex,20)=(reg_0(j,12))/clients+feederoutput(i+feederindex,20);

regdisoutput(i+feederindex,20)=(reg_0(j,12))+regdisoutput(i+feederindex,20);

            end
            if reg_0(j,1)>=37622 && reg_0(j,1)<=37986
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end
               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step

                if saididaily<feederMEDtuki(i+feederindex,9)&&(clients>50)

feederoutputMEDDED(i+feederindex,21)=(reg_0(j,12))/clients+feederoutputMEDDED(i+feederindex,21);

regdisoutputMEDDED(i+feederindex,21)=(reg_0(j,12))+regdisoutputMEDDED(i+feederindex,21);
                end

feederoutput(i+feederindex,21)=(reg_0(j,12))/clients+feederoutput(i+feederindex,21);

regdisoutput(i+feederindex,21)=(reg_0(j,12))+regdisoutput(i+feederindex,21);

            end
            if reg_0(j,1)>=37987 && reg_0(j,1)<=38352
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end
               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;
```

```
saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step

                    if saididaily<feederMEDtuki(i+feederindex,10)&&(clients>50)

feederoutputMEDDED(i+feederindex,22)=(reg_0(j,12))/clients+feederoutputMEDDED(i+feederindex,22);

regdisoutputMEDDED(i+feederindex,22)=(reg_0(j,12))+regdisoutputMEDDED(i+feederindex,22);
                    end

feederoutput(i+feederindex,22)=(reg_0(j,12))/clients+feederoutput(i+feederindex,22);

regdisoutput(i+feederindex,22)=(reg_0(j,12))+regdisoutput(i+feederindex,22);

                end

              end
            end

          end

        end
      if i==feedernames_size(1,1) %If added to fix the problem when the last feeder was read on feedernames, which had no next feeder to use as end index in the previous IF.

          for j=feedernames(feedernames_size(1,1),2):reg_0_size(1,1)

            if ((reg_0(j,7)>momentary)&&(reg_0(j,2)~=reg_0(j-1,2)))

              if reg_0(j,1)>=35796 && reg_0(j,1)<=36160
                if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
                else
                  clients=feederoutputMEDDED(i+feederindex,4);
                end
                if clients==0
                  clients=99;
                end
                saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step
```

```
                if (saididaily<feederMEDtuki(i+feederindex,4))&&(r~=4)&&(clients>50)

feederoutputMEDDED(i+feederindex,16)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,16);

regdisoutputMEDDED(i+feederindex,16)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,16);
                end

feederoutput(i+feederindex,16)=(reg_0(j,12))/clients+feederoutput(i+feederindex,16);

regdisoutput(i+feederindex,16)=(reg_0(j,12))+regdisoutput(i+feederindex,16);
            end

            if reg_0(j,1)>=36161 && reg_0(j,1)<=36525
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end
               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step

                if saididaily<feederMEDtuki(i+feederindex,5)&&(clients>50)

feederoutputMEDDED(i+feederindex,17)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,17);

regdisoutputMEDDED(i+feederindex,17)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,17);
                end

feederoutput(i+feederindex,17)=(reg_0(j,12))/clients+feederoutput(i+feederindex,17);

regdisoutput(i+feederindex,17)=(reg_0(j,12))+regdisoutput(i+feederindex,17);
            end

            if reg_0(j,1)>=36526 && reg_0(j,1)<=36891
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
```

```
                clients=reg_0(j,12);
            else
                clients=feederoutputMEDDED(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end
            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step

            if saididaily<feederMEDtuki(i+feederindex,6)&&(clients>50)

feederoutputMEDDED(i+feederindex,18)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,18);

regdisoutputMEDDED(i+feederindex,18)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,18);
            end

feederoutput(i+feederindex,18)=(reg_0(j,12))/clients+feederoutput(i+feederindex,18);

regdisoutput(i+feederindex,18)=(reg_0(j,12))+regdisoutput(i+feederindex,18);
        end

        if reg_0(j,1)>=36892 && reg_0(j,1)<=37256
            if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                clients=reg_0(j,12);
            else
                clients=feederoutputMEDDED(i+feederindex,4);
            end
            if clients==0
                clients=99;
            end
            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step

            if saididaily<feederMEDtuki(i+feederindex,7)&&(clients>50)

feederoutputMEDDED(i+feederindex,19)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,19);
```

```
regdisoutputMEDDED(i+feederindex,19)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,19);
                end

feederoutput(i+feederindex,19)=(reg_0(j,12))/clients+feederoutput(i+feederindex,19);

regdisoutput(i+feederindex,19)=(reg_0(j,12))+regdisoutput(i+feederindex,19);
            end

            if reg_0(j,1)>=37257 && reg_0(j,1)<=37621
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
               end
               if clients==0
                  clients=99;
               end
               saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),re
g_0(j,13),inputoption); %Adjustment step

               if saididaily<feederMEDtuki(i+feederindex,8)&&(clients>50)

feederoutputMEDDED(i+feederindex,20)=(reg_0(j,12))/clients+feederoutputMEDDED(i
+feederindex,20);

regdisoutputMEDDED(i+feederindex,20)=(reg_0(j,12))+regdisoutputMEDDED(i+feeder
index,20);
                end

feederoutput(i+feederindex,20)=(reg_0(j,12))/clients+feederoutput(i+feederindex,20);

regdisoutput(i+feederindex,20)=(reg_0(j,12))+regdisoutput(i+feederindex,20);
            end

            if reg_0(j,1)>=37622 && reg_0(j,1)<=37986
               if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                  clients=reg_0(j,12);
               else
                  clients=feederoutputMEDDED(i+feederindex,4);
               end
               if clients==0
```

```
                clients=99;
            end
            saididaily=(reg_0(j,12)*reg_0(j,7))/clients;

saididaily=saididailyWEIGHTER3(saididaily,feederoutputMEDDED(i+feederindex,2),reg_0(j,13),inputoption); %Adjustment step

            if saididaily<feederMEDtuki(i+feederindex,9)&&(clients>50)

feederoutputMEDDED(i+feederindex,21)=(reg_0(j,12))/clients+feederoutputMEDDED(i+feederindex,21);

regdisoutputMEDDED(i+feederindex,21)=(reg_0(j,12))+regdisoutputMEDDED(i+feederindex,21);
            end

feederoutput(i+feederindex,21)=(reg_0(j,12))/clients+feederoutput(i+feederindex,21);

regdisoutput(i+feederindex,21)=(reg_0(j,12))+regdisoutput(i+feederindex,21);

        end
        if reg_0(j,1)>=37987 && reg_0(j,1)<=38352
            if reg_0(j,12)>feederoutputMEDDED(i+feederindex,4)
                clients=reg_0(j,12);
            else
                clients=feederoutputMEDDED(i+feederindex,4);
            end
            if clients==0
                clients=99;
                endsaididaily=(reg_0(j,12)*reg_0(j,7))/clients;
                if saididaily<feederMEDtuki(i+feederindex,10)&&(clients>50)

feederoutputMEDDED(i+feederindex,22)=(reg_0(j,12))/clients+feederoutputMEDDED(i+feederindex,22);

regdisoutputMEDDED(i+feederindex,22)=(reg_0(j,12))+regdisoutputMEDDED(i+feederindex,22);
                end

feederoutput(i+feederindex,22)=(reg_0(j,12))/clients+feederoutput(i+feederindex,22);

regdisoutput(i+feederindex,22)=(reg_0(j,12))+regdisoutput(i+feederindex,22);
            end
        end
```

```
        end
      end

   end
   if feederoutputMEDDED(i+feederindex,4)==0
      feederoutputMEDDED(i+feederindex,16)=999;
      feederoutputMEDDED(i+feederindex,17)=999;
      feederoutputMEDDED(i+feederindex,18)=999;
      feederoutputMEDDED(i+feederindex,19)=999;
      feederoutputMEDDED(i+feederindex,20)=999;
      feederoutputMEDDED(i+feederindex,21)=999;
      feederoutputMEDDED(i+feederindex,22)=999;
      feederoutput(i+feederindex,16)=999;
      feederoutput(i+feederindex,17)=999;
      feederoutput(i+feederindex,18)=999;
      feederoutput(i+feederindex,19)=999;
      feederoutput(i+feederindex,20)=999;
      feederoutput(i+feederindex,21)=999;
      feederoutput(i+feederindex,22)=999;
   end

end

feederoutputMEDDED_size=size(feederoutputMEDDED);
feederindex=feederindex+feedernames_size(1,1);

if r==0
   reg0_saididaily=saididailytemp;
   reg0_saididailyMEDDED=saididailytempMEDDED;
end
if r==1
   reg1_saididaily=saididailytemp;
   reg1_saididailyMEDDED=saididailytempMEDDED;
end
if r==2
   reg2_saididaily=saididailytemp;
   reg2_saididailyMEDDED=saididailytempMEDDED;
end
if r==3
   reg3_saididaily=saididailytemp;
   reg3_saididailyMEDDED=saididailytempMEDDED;
end
if r==4
   reg4_saididaily=saididailytemp;
   reg4_saididailyMEDDED=saididailytempMEDDED;
```

```
   end
   if r==5
      reg5_saididaily=saididailytemp;
      reg5_saididailyMEDDED=saididailytempMEDDED;
   end
   if r==6
      reg6_saididaily=saididailytemp;
      reg6_saididailyMEDDED=saididailytempMEDDED;
   end

end



%-------------------------------------------------------------------------%
%//////////////////////END OF MEDDED SAIDI SAIFI//////////////////////////%
%-------------------------------------------------------------------------%



save outputs_adjusted.mat feederoutput feederoutputMEDDED regdisoutput
regdisoutputMEDDED

counter=1;
killed=0;
for k=1:(feederoutput_size)

   if feederoutput(counter,4)==0
      feederoutput(counter,:)=[];
      feederMEDtuki(counter,:)=[];
      feederoutputMEDDED(counter,:)=[];

      regdisoutput(counter,:)=[];
      regdisoutputMEDDED(counter,:)=[];
      killed=killed+1;
      counter=counter-1;
   end
   counter=counter+1;
end
feederoutput_size=size(feederoutput);
feederMEDtuki_size=size(feederMEDtuki);
feederoutputMEDDED_size=size(feederoutputMEDDED);


%-------------------------------------------------------------------------
```

```
%//////////////////////////CONGLOMERATOR////////////////////////////////
%-------------------------------------------------------------------------

systemoutput=zeros(1,17);


%----Conglomerate by Region

regionoutput=zeros(7,17);
regionoutputMEDDED=zeros(7,17);


index=1;
for regnum=[0,1,2,3,4,5,6]
   for i=1:feederoutput_size(1,1)

      if feederoutput(i,1)==regnum
         regionoutput(index,1)=regnum;

         %SAIDI
         regionoutput(index,3)=regionoutput(index,3)+regdisoutput(i,8); %98
         regionoutput(index,4)=regionoutput(index,4)+regdisoutput(i,9); %99
         regionoutput(index,5)=regionoutput(index,5)+regdisoutput(i,10); %00
         regionoutput(index,6)=regionoutput(index,6)+regdisoutput(i,11); %01
         regionoutput(index,7)=regionoutput(index,7)+regdisoutput(i,12); %02
         regionoutput(index,8)=regionoutput(index,8)+regdisoutput(i,13); %03
         regionoutput(index,9)=regionoutput(index,9)+regdisoutput(i,14); %04

         %SAIFI
         regionoutput(index,11)=regionoutput(index,11)+regdisoutput(i,16); %98
         regionoutput(index,12)=regionoutput(index,12)+regdisoutput(i,17); %99
         regionoutput(index,13)=regionoutput(index,13)+regdisoutput(i,18); %00
         regionoutput(index,14)=regionoutput(index,14)+regdisoutput(i,19); %01
         regionoutput(index,15)=regionoutput(index,15)+regdisoutput(i,20); %02
         regionoutput(index,16)=regionoutput(index,16)+regdisoutput(i,21); %03
         regionoutput(index,17)=regionoutput(index,17)+regdisoutput(i,22); %04


         %SAIDI MEDDED

regionoutputMEDDED(index,3)=regionoutputMEDDED(index,3)+regdisoutputMEDDE
D(i,8); %98

regionoutputMEDDED(index,4)=regionoutputMEDDED(index,4)+regdisoutputMEDDE
D(i,9); %99
```

```
regionoutputMEDDED(index,5)=regionoutputMEDDED(index,5)+regdisoutputMEDDE
D(i,10); %00

regionoutputMEDDED(index,6)=regionoutputMEDDED(index,6)+regdisoutputMEDDE
D(i,11); %01

regionoutputMEDDED(index,7)=regionoutputMEDDED(index,7)+regdisoutputMEDDE
D(i,12); %02

regionoutputMEDDED(index,8)=regionoutputMEDDED(index,8)+regdisoutputMEDDE
D(i,13); %03

regionoutputMEDDED(index,9)=regionoutputMEDDED(index,9)+regdisoutputMEDDE
D(i,14); %04


        %SAIFI MEDDED

regionoutputMEDDED(index,11)=regionoutputMEDDED(index,11)+regdisoutputMED
DED(i,16); %98

regionoutputMEDDED(index,12)=regionoutputMEDDED(index,12)+regdisoutputMED
DED(i,17); %99

regionoutputMEDDED(index,13)=regionoutputMEDDED(index,13)+regdisoutputMED
DED(i,18); %00

regionoutputMEDDED(index,14)=regionoutputMEDDED(index,14)+regdisoutputMED
DED(i,19); %01

regionoutputMEDDED(index,15)=regionoutputMEDDED(index,15)+regdisoutputMED
DED(i,20); %02

regionoutputMEDDED(index,16)=regionoutputMEDDED(index,16)+regdisoutputMED
DED(i,21); %03

regionoutputMEDDED(index,17)=regionoutputMEDDED(index,17)+regdisoutputMED
DED(i,22); %04


      end

   end
```

```
    index=index+1;


end

systemoutput=(regionoutput(1,:)+regionoutput(2,:)+regionoutput(3,:)+regionoutput(4,:)+
regionoutput(5,:)+regionoutput(6,:)+regionoutput(7,:))/systemclients;
systemoutputMEDDED=(regionoutputMEDDED(1,:)+regionoutputMEDDED(2,:)+regio
noutputMEDDED(3,:)+regionoutputMEDDED(4,:)+regionoutputMEDDED(5,:)+regiono
utputMEDDED(6,:)+regionoutputMEDDED(7,:))/systemclients;

tukitu=regionoutput;
tukituMEDDED=regionoutputMEDDED;

regionoutput(1,:)=regionoutput(1,:)/regionclients(1,2);
regionoutput(2,:)=regionoutput(2,:)/regionclients(2,2);
regionoutput(3,:)=regionoutput(3,:)/regionclients(3,2);
regionoutput(4,:)=regionoutput(4,:)/regionclients(4,2);
regionoutput(5,:)=regionoutput(5,:)/regionclients(5,2);
regionoutput(6,:)=regionoutput(6,:)/regionclients(6,2);
regionoutput(7,:)=regionoutput(7,:)/regionclients(7,2);

regionoutputMEDDED(1,:)=regionoutputMEDDED(1,:)/regionclients(1,2);
regionoutputMEDDED(2,:)=regionoutputMEDDED(2,:)/regionclients(2,2);
regionoutputMEDDED(3,:)=regionoutputMEDDED(3,:)/regionclients(3,2);
regionoutputMEDDED(4,:)=regionoutputMEDDED(4,:)/regionclients(4,2);
regionoutputMEDDED(5,:)=regionoutputMEDDED(5,:)/regionclients(5,2);
regionoutputMEDDED(6,:)=regionoutputMEDDED(6,:)/regionclients(6,2);
regionoutputMEDDED(7,:)=regionoutputMEDDED(7,:)/regionclients(7,2);

regionoutput(1,1)=0;
regionoutput(2,1)=1;
regionoutput(3,1)=2;
regionoutput(4,1)=3;
regionoutput(5,1)=4;
regionoutput(6,1)=5;
regionoutput(7,1)=6;

regionoutputMEDDED(1,1)=0;
regionoutputMEDDED(2,1)=1;
regionoutputMEDDED(3,1)=2;
regionoutputMEDDED(4,1)=3;
regionoutputMEDDED(5,1)=4;
regionoutputMEDDED(6,1)=5;
regionoutputMEDDED(7,1)=6;
```

```
%-----Conglomerate Data by District----------------------------------------

districtoutput=zeros(27,18); %FORMAT: 1-Region 2-District 3-ZERO 4-10-SAIDI 11-
ZERO 12-19-SAIFI
districtoutputMEDDED=zeros(27,18);
index=1;
for
disnum=[20,22,23,24,95,10,11,12,13,14,34,40,42,44,45,50,53,54,55,60,61,62,63,70,71,7
2,73]
   for i=1:feederoutput_size(1,1)

      if regdisoutput(i,2)==disnum

         %SAIDI

         districtoutput(index,2)=disnum;
         districtoutput(index,3)=districtoutput(index,3)+regdisoutput(i,8); %98
         districtoutput(index,4)=districtoutput(index,4)+regdisoutput(i,9); %99
         districtoutput(index,5)=districtoutput(index,5)+regdisoutput(i,10);%00
         districtoutput(index,6)=districtoutput(index,6)+regdisoutput(i,11);%01
         districtoutput(index,7)=districtoutput(index,7)+regdisoutput(i,12);%02
         districtoutput(index,8)=districtoutput(index,8)+regdisoutput(i,13);%03
         districtoutput(index,9)=districtoutput(index,9)+regdisoutput(i,14);%04

         %SAIFI

         districtoutput(index,11)=districtoutput(index,11)+regdisoutput(i,16);%98
         districtoutput(index,12)=districtoutput(index,12)+regdisoutput(i,17);%99
         districtoutput(index,13)=districtoutput(index,13)+regdisoutput(i,18);%00
         districtoutput(index,14)=districtoutput(index,14)+regdisoutput(i,19);%01
         districtoutput(index,15)=districtoutput(index,15)+regdisoutput(i,20);%02
         districtoutput(index,16)=districtoutput(index,16)+regdisoutput(i,21);%03
         districtoutput(index,17)=districtoutput(index,17)+regdisoutput(i,22);%04

         %SAIDI MEDDED

         districtoutputMEDDED(index,2)=disnum;

districtoutputMEDDED(index,3)=districtoutputMEDDED(index,3)+regdisoutputMEDD
ED(i,8); %98
```

```
districtoutputMEDDED(index,4)=districtoutputMEDDED(index,4)+regdisoutputMEDD
ED(i,9); %99

districtoutputMEDDED(index,5)=districtoutputMEDDED(index,5)+regdisoutputMEDD
ED(i,10);%00

districtoutputMEDDED(index,6)=districtoutputMEDDED(index,6)+regdisoutputMEDD
ED(i,11);%01

districtoutputMEDDED(index,7)=districtoutputMEDDED(index,7)+regdisoutputMEDD
ED(i,12);%02

districtoutputMEDDED(index,8)=districtoutputMEDDED(index,8)+regdisoutputMEDD
ED(i,13);%03

districtoutputMEDDED(index,9)=districtoutputMEDDED(index,9)+regdisoutputMEDD
ED(i,14);%04

        %SAIFI MEDDED


districtoutputMEDDED(index,11)=districtoutputMEDDED(index,11)+regdisoutputMED
DED(i,16);%98

districtoutputMEDDED(index,12)=districtoutputMEDDED(index,12)+regdisoutputMED
DED(i,17);%99

districtoutputMEDDED(index,13)=districtoutputMEDDED(index,13)+regdisoutputMED
DED(i,18);%00

districtoutputMEDDED(index,14)=districtoutputMEDDED(index,14)+regdisoutputMED
DED(i,19);%01

districtoutputMEDDED(index,15)=districtoutputMEDDED(index,15)+regdisoutputMED
DED(i,20);%02

districtoutputMEDDED(index,16)=districtoutputMEDDED(index,16)+regdisoutputMED
DED(i,21);%03

districtoutputMEDDED(index,17)=districtoutputMEDDED(index,17)+regdisoutputMED
DED(i,22);%04

      end
```

```
   end
   index=index+1;


end

for i=1:27
   for j=3:18

      districtoutput(i,j)=districtoutput(i,j)/districtclients(i,2);

   end
end

for i=1:27
   for j=3:18

      districtoutputMEDDED(i,j)=districtoutputMEDDED(i,j)/districtclients(i,2);

   end
end



%feeder_list %Feederclients column one. used as index for for. other columns
eliminated.

for i=1:feederclients_size(1,1)

   for j=1:feederoutput_size(1,1)

      if feederclients(i,1)==feederoutput(j,3)

         feederclients_append(i,2)=feederclients_append(i,1)+feederoutput(j,8);
         feederclients_append(i,3)=feederclients_append(i,3)+feederoutput(j,9);
         feederclients_append(i,4)=feederclients_append(i,4)+feederoutput(j,10);
         feederclients_append(i,5)=feederclients_append(i,5)+feederoutput(j,11);
         feederclients_append(i,6)=feederclients_append(i,6)+feederoutput(j,12);
         feederclients_append(i,7)=feederclients_append(i,7)+feederoutput(j,13);
         feederclients_append(i,8)=feederclients_append(i,8)+feederoutput(j,14);

         feederclients_append(i,10)=feederclients_append(i,10)+feederoutput(j,16);
         feederclients_append(i,11)=feederclients_append(i,11)+feederoutput(j,17);
         feederclients_append(i,12)=feederclients_append(i,12)+feederoutput(j,18);
         feederclients_append(i,13)=feederclients_append(i,13)+feederoutput(j,19);
```

```
feederclients_append(i,14)=feederclients_append(i,14)+feederoutput(j,20);
feederclients_append(i,15)=feederclients_append(i,15)+feederoutput(j,21);
feederclients_append(i,16)=feederclients_append(i,16)+feederoutput(j,22);


feederclients_append_MEDDED(i,2)=feederclients_append_MEDDED(i,2)+feederoutpu
t(j,8);

feederclients_append_MEDDED(i,3)=feederclients_append_MEDDED(i,3)+feederoutpu
t(j,9);

feederclients_append_MEDDED(i,4)=feederclients_append_MEDDED(i,4)+feederoutpu
t(j,10);

feederclients_append_MEDDED(i,5)=feederclients_append_MEDDED(i,5)+feederoutpu
t(j,11);

feederclients_append_MEDDED(i,6)=feederclients_append_MEDDED(i,6)+feederoutpu
t(j,12);

feederclients_append_MEDDED(i,7)=feederclients_append_MEDDED(i,7)+feederoutpu
t(j,13);

feederclients_append_MEDDED(i,8)=feederclients_append_MEDDED(i,8)+feederoutpu
t(j,14);


feederclients_append_MEDDED(i,10)=feederclients_append_MEDDED(i,10)+feederout
put(j,16);

feederclients_append_MEDDED(i,11)=feederclients_append_MEDDED(i,11)+feederout
put(j,17);

feederclients_append_MEDDED(i,12)=feederclients_append_MEDDED(i,12)+feederout
put(j,18);

feederclients_append_MEDDED(i,13)=feederclients_append_MEDDED(i,13)+feederout
put(j,19);

feederclients_append_MEDDED(i,14)=feederclients_append_MEDDED(i,14)+feederout
put(j,20);

feederclients_append_MEDDED(i,15)=feederclients_append_MEDDED(i,15)+feederout
put(j,21);
```

```
feederclients_append_MEDDED(i,16)=feederclients_append_MEDDED(i,16)+feederout
put(j,22);

      end
   end
end

feederclients_total=horzcat(feederclients,feederclients_append);
feederclients_total_MEDDED=horzcat(feederclients,feederclients_append_MEDDED);


if inputoption==1

save saididailys_adjusted_op1.mat *saididaily *saididailyMEDDED
save feederoutput_adjusted_op1.txt feederoutput -ascii
save feederoutputMEDDED_adjusted_op1.txt feederoutputMEDDED -ascii
save regionoutput_adjusted_op1.txt regionoutput -ascii
save regionoutputMEDDED_adjusted_op1.txt regionoutputMEDDED -ascii
save districtoutput_adjusted_op1.txt districtoutput -ascii
save districtoutputMEDDED_adjusted_op1.txt districtoutputMEDDED -ascii

elseif inputoption==2

   save saididailys_adjusted_op2.mat *saididaily *saididailyMEDDED
save feederoutput_adjusted_op2.txt feederoutput -ascii
save feederoutputMEDDED_adjusted_op2.txt feederoutputMEDDED -ascii
save regionoutput_adjusted_op2.txt regionoutput -ascii
save regionoutputMEDDED_adjusted_op2.txt regionoutputMEDDED -ascii
save districtoutput_adjusted_op2.txt districtoutput -ascii
save districtoutputMEDDED_adjusted_op2.txt districtoutputMEDDED -ascii

elseif inputoption==3

   save saididailys_adjusted_op3.mat *saididaily *saididailyMEDDED
save feederoutput_adjusted_op3.txt feederoutput -ascii
save feederoutputMEDDED_adjusted_op3.txt feederoutputMEDDED -ascii
save regionoutput_adjusted_op3.txt regionoutput -ascii
save regionoutputMEDDED_adjusted_op3.txt regionoutputMEDDED -ascii
save districtoutput_adjusted_op3.txt districtoutput -ascii
save districtoutputMEDDED_adjusted_op3.txt districtoutputMEDDED -ascii

elseif inputoption==4

   save saididailys_adjusted_op4.mat *saididaily *saididailyMEDDED
```

```
save feederoutput_adjusted_op4.txt feederoutput -ascii
save feederoutputMEDDED_adjusted_op4.txt feederoutputMEDDED -ascii
save regionoutput_adjusted_op4.txt regionoutput -ascii
save regionoutputMEDDED_adjusted_op4.txt regionoutputMEDDED -ascii
save districtoutput_adjusted_op4.txt districtoutput -ascii
save districtoutputMEDDED_adjusted_op4.txt districtoutputMEDDED -ascii

end


toc
```

# APPENDIX C: WEIGHTER SCRIPT

## Script for Preparing Data for WEIGHTER:

```
%New file Containing all the saididaily scripts in one. It includes the
%functions of:
%CLEANER-eliminates 0 data of lists (regionwise)
%YEARLY-organizes data in yearly datasets (regionwise)
%BREAKER2-makes yearly data for districts from region data of YEARLY.

%This updated version of the scripts will work for both RIPPER and STRIPPER
%This file will work with non-medded saididailys.

clear all

EXVX_Uave=zeros(2,1);

select=input('RIPPER (1) or STRIPPER (2) INPUT? ');
option=input('OPTION 1-2-3-4? ');
if select==1

   index=1;
elseif select==2
   index=3

end


tic


%for option=1:index

   if select==1

      load saididailys.mat

   elseif select==2 && option==1

      load saididailys_adjusted_op1.mat


   elseif select==2 && option==2
```

```
    load saididailys_adjusted_op2.mat

elseif select==2 && option==3

    load saididailys_adjusted_op3.mat

elseif select==2 && option==4

    load saididailys_adjusted_op4.mat

end

%-----CLEANER PROCESS
for r=0:6
    if r==0
        saididailytemp=reg0_saididaily;
        sizetemp=size(saididailytemp);
    end
    if r==1
        saididailytemp=reg1_saididaily;
        sizetemp=size(saididailytemp);
    end
    if r==2
        saididailytemp=reg2_saididaily;
        sizetemp=size(saididailytemp);
    end
    if r==3
        saididailytemp=reg3_saididaily;
        sizetemp=size(saididailytemp);
    end
    if r==4
        saididailytemp=reg4_saididaily;
        sizetemp=size(saididailytemp);
    end
    if r==5
        saididailytemp=reg5_saididaily;
        sizetemp=size(saididailytemp);
    end
    if r==6
        saididailytemp=reg6_saididaily;
        sizetemp=size(saididailytemp);
    end
    k=1;
    for i=1:sizetemp(1,1)
```

```
        if saididailytemp(k,10)==0 && saididailytemp(k,4)==0 && saididailytemp(k,5)==0 && saididailytemp(k,6)==0 && saididailytemp(k,7)==0 && saididailytemp(k,8)==0 && saididailytemp(k,9)==0

            saididailytemp(k,:)=[];
            k=k-1;
        end
        k=k+1;
    end


    if r==0
        reg0_saididaily=saididailytemp;
        %save saididailys2.mat *saididaily
        p=0

    end
    if r==1
        reg1_saididaily=saididailytemp;
        %save saididailys2.mat *saididaily
        p=1

    end
    if r==2
        reg2_saididaily=saididailytemp;
        %save saididailys2.mat *saididaily
        p=2

    end
    if r==3
        reg3_saididaily=saididailytemp;
        %save saididailys2.mat *saididaily
        p=3

    end
    if r==4
        reg4_saididaily=saididailytemp;
        %save saididailys2.mat *saididaily
        p=4

    end
    if r==5
        reg5_saididaily=saididailytemp;
        %save saididailys2.mat *saididaily
        p=5
```

```
    end
    if r==6
        reg6_saididaily=saididailytemp;
        %save saididailys2.mat *saididaily
        p=6

    end

end

%-----YEARLY PROCESS

saididailyyearly=zeros(1,10);
systemclients=1537306;
a=1;
b=1;
c=1;
d=1;
e=1;
f=1;
g=1;

for r=0:6

    if r==0
        saididailytemp=reg0_saididaily;
        sizetemp=size(saididailytemp);
        p=0
    end
    if r==1
        saididailytemp=reg1_saididaily;
        sizetemp=size(saididailytemp);
        p=1
    end
    if r==2
        saididailytemp=reg2_saididaily;
        sizetemp=size(saididailytemp);
        p=2
    end
    if r==3
        saididailytemp=reg3_saididaily;
        sizetemp=size(saididailytemp);
        p=3
    end
```

```
if r==4
   saididailytemp=reg4_saididaily;
   sizetemp=size(saididailytemp);
   p=4
end
if r==5
   saididailytemp=reg5_saididaily;
   sizetemp=size(saididailytemp);
   p=5
end
if r==6
   saididailytemp=reg6_saididaily;
   sizetemp=size(saididailytemp);
   p=6
end

for i=1:sizetemp(1,1)

   if saididailytemp(i,4)>0

      saididailyyearly(a,4)=saididailytemp(i,4);
      a=a+1;

   elseif saididailytemp(i,5)>0

      saididailyyearly(b,5)=saididailytemp(i,5);
      b=b+1;

   elseif saididailytemp(i,6)>0

      saididailyyearly(c,6)=saididailytemp(i,6);
      c=c+1;

   elseif saididailytemp(i,7)>0

      saididailyyearly(d,7)=saididailytemp(i,7);
      d=d+1;

   elseif saididailytemp(i,8)>0

      saididailyyearly(e,8)=saididailytemp(i,8);
      e=e+1;

   elseif saididailytemp(i,9)>0
```

```
        saididailyyearly(f,9)=saididailytemp(i,9);
        f=f+1;

    elseif saididailytemp(i,10)>0

        saididailyyearly(g,10)=saididailytemp(i,10);
        g=g+1;

    end


  end



end



saididaily1998=sortrows(saididailyyearly(:,4),-1);
saididaily1999=sortrows(saididailyyearly(:,5),-1);
saididaily2000=sortrows(saididailyyearly(:,6),-1);
saididaily2001=sortrows(saididailyyearly(:,7),-1);
saididaily2002=sortrows(saididailyyearly(:,8),-1);
saididaily2003=sortrows(saididailyyearly(:,9),-1);
saididaily2004=sortrows(saididailyyearly(:,10),-1);


for r=0:6

  if r==0
    saididailytemp=saididaily1998;
    sizetemp=size(saididailytemp);

  end
  if r==1
    saididailytemp=saididaily1999;
    sizetemp=size(saididailytemp);

  end
  if r==2
    saididailytemp=saididaily2000;
    sizetemp=size(saididailytemp);

  end
```

```
if r==3
   saididailytemp=saididaily2001;
   sizetemp=size(saididailytemp);

end
if r==4
   saididailytemp=saididaily2002;
   sizetemp=size(saididailytemp);

end
if r==5
   saididailytemp=saididaily2003;
   sizetemp=size(saididailytemp);

end
if r==6
   saididailytemp=saididaily2004;
   sizetemp=size(saididailytemp);

end

k=1;
for i=1:sizetemp(1,1)

   if saididailytemp(k,1)==0

      saididailytemp(k,:)=[];
      k=k-1;
   end
   k=k+1;
end

if r==0
   saididaily1998=saididailytemp/systemclients;
   saididaily1998log=log(saididaily1998);
   theta1998=mean(saididaily1998log); %theta = mean in Normal mode
   dev1998=std(saididaily1998log);    %omega = variance in Normal mode
   omega1998=dev1998^2;
   EofX1998=exp((theta1998+omega1998/2));
   VofX1998=exp((2*theta1998+omega1998))*(exp(omega1998)-1);
end
if r==1
   saididaily1999=saididailytemp/systemclients;
   saididaily1999log=log(saididaily1999);
   theta1999=mean(saididaily1999log);
```

```
   dev1999=std(saididaily1999log);
   omega1999=dev1999^2;
   EofX1999=exp((theta1999+omega1999/2));
   VofX1999=exp((2*theta1999+omega1999))*(exp(omega1999)-1);
end
if r==2
   saididaily2000=saididailytemp/systemclients;
   saididaily2000log=log(saididaily2000);
   theta2000=mean(saididaily2000log);
   dev2000=std(saididaily2000log);
   omega2000=dev2000^2;
   EofX2000=exp((theta2000+omega2000/2));
   VofX2000=exp((2*theta2000+omega2000))*(exp(omega2000)-1);
end
if r==3
   saididaily2001=saididailytemp/systemclients;
   saididaily2001log=log(saididaily2001);
   theta2001=mean(saididaily2001log);
   dev2001=std(saididaily2001log);
   omega2001=dev2001^2;
   EofX2001=exp((theta2001+omega2001/2));
   VofX2001=exp((2*theta2001+omega2001))*(exp(omega2001)-1);
end
if r==4
   saididaily2002=saididailytemp/systemclients;
   saididaily2002log=log(saididaily2002);
   theta2002=mean(saididaily2002log);
   dev2002=std(saididaily2002log);
   omega2002=dev2002^2;
   EofX2002=exp((theta2002+omega2002/2));
   VofX2002=exp((2*theta2002+omega2002))*(exp(omega2002)-1);
end
if r==5
   saididaily2003=saididailytemp/systemclients;
   saididaily2003log=log(saididaily2003);
   theta2003=mean(saididaily2003log);
   dev2003=std(saididaily2003log);
   omega2003=dev2003^2;
   EofX2003=exp((theta2003+omega2003/2));
   VofX2003=exp((2*theta2003+omega2003))*(exp(omega2003)-1);
end
if r==6
   saididaily2004=saididailytemp/systemclients;
   saididaily2004log=log(saididaily2004);
   theta2004=mean(saididaily2004log);
```

```
        dev2004=std(saididaily2004log);
        omega2004=dev2004^2;
        EofX2004=exp((theta2004+omega2004/2));
        VofX2004=exp((2*theta2004+omega2004))*(exp(omega2004)-1);
      end
    end

    %--BREAKER2 PROCESS



    header1=[0,0,0,0,0,0,99,1,1,1,1,99,2,2,2,2,99,3,3,3,3,99,4,4,4,4,99,5,5,5,5,99,6,6,6,6];

header2=[20,22,23,24,95,0,10,11,12,0,13,14,34,0,40,42,44,45,0,50,53,54,55,0,60,61,62,6
3,0,70,71,72,73];
    header4=[0,1,2,3,4,5,6];

    saididaily_district_numonly(1,:)=header1;
    saididaily_district_numonly(2,:)=header2;
    saididaily_district_div=saididaily_district_numonly;
    saididailytemplong=zeros(1,1);
    load districtclients2.mat
    load regiones.mat regionclients

    EXVX_district=zeros(4,33);
    EXVX_district(1,:)=header1;
    EXVX_district(2,:)=header2;

    EXVX_region=zeros(3,7);
    EXVX_region(1,:)=header4;



    a=3;
    b=3;
    c=3;
    d=3;
    e=3;
    f=3;
    g=3;
    h=3;

    j=3;
    k=3;
    l=3;
```

```
m=3;
n=3;
o=3;
p=3;
q=3;

s=3;
t=3;
u=3;
v=3;
w=3;
x=3;
y=3;
z=3;
aa=3;
bb=3;
cc=3;
dd=3;
ee=3;
ff=3;
gg=3;

for r=0:6

   if r==0
      saididailytemp=reg0_saididaily;
      sizetemp=size(saididailytemp);

   end
   if r==1
      saididailytemp=reg1_saididaily;
      sizetemp=size(saididailytemp);

   end
   if r==2
      saididailytemp=reg2_saididaily;
      sizetemp=size(saididailytemp);

   end
   if r==3
      saididailytemp=reg3_saididaily;
      sizetemp=size(saididailytemp);

   end
   if r==4
```

```
    saididailytemp=reg4_saididaily;
    sizetemp=size(saididailytemp);

end
if r==5
    saididailytemp=reg5_saididaily;
    sizetemp=size(saididailytemp);

end
if r==6
    saididailytemp=reg6_saididaily;
    sizetemp=size(saididailytemp);

end

for i=1:sizetemp(1,1)

    if saididailytemp(i,2)==20 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(a,1)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp
(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(a,1)=saididaily_district_numonly(a,1)/districtclients2(2,1);
        a=a+1;


    elseif saididailytemp(i,2)==22 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(b,2)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp
(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(b,2)=saididaily_district_numonly(b,2)/districtclients2(2,2);
        b=b+1;

    elseif saididailytemp(i,2)==23 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0
```

```
saididaily_district_numonly(c,3)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp
(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(c,3)=saididaily_district_numonly(c,3)/districtclients2(2,3);
            c=c+1;

        elseif                  saididailytemp(i,2)==24                 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(d,4)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp
(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(d,4)=saididaily_district_numonly(d,4)/districtclients2(2,4);
            d=d+1;

        elseif                  saididailytemp(i,2)==95                 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(e,5)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp
(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(e,5)=saididaily_district_numonly(e,5)/districtclients2(2,5);
            e=e+1;

        elseif                  saididailytemp(i,2)==10                 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(f,7)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp
(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(f,7)=saididaily_district_numonly(f,7)/districtclients2(2,7);
            f=f+1;

        elseif                  saididailytemp(i,2)==11                 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0
```

```
saididaily_district_numonly(g,8)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp
(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(g,8)=saididaily_district_numonly(g,8)/districtclients2(2,8);
            g=g+1;

        elseif                  saididailytemp(i,2)==12                 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(h,9)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp
(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(h,9)=saididaily_district_numonly(h,9)/districtclients2(2,9);
            h=h+1;

        elseif                  saididailytemp(i,2)==13                 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(j,11)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytem
p(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(j,11)=saididaily_district_numonly(j,11)/districtclients2(2,11);
            j=j+1;

        elseif                  saididailytemp(i,2)==14                 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(k,12)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytem
p(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(k,12)=saididaily_district_numonly(k,12)/districtclients2(2,12);
            k=k+1;

        elseif                  saididailytemp(i,2)==34                 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0
```

```
saididaily_district_numonly(l,13)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytem
p(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(l,13)=saididaily_district_numonly(l,13)/districtclients2(2,13);
            l=l+1;
        elseif                  saididailytemp(i,2)==40                 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(m,15)=saididailytemp(i,4)+saididailytemp(i,5)+saididailyte
mp(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10
);

saididaily_district_div(m,15)=saididaily_district_numonly(m,15)/districtclients2(2,15);
            m=m+1;

        elseif                  saididailytemp(i,2)==42                 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(n,16)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytem
p(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(n,16)=saididaily_district_numonly(n,16)/districtclients2(2,16);
            n=n+1;

        elseif                  saididailytemp(i,2)==44                 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(o,17)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytem
p(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(o,17)=saididaily_district_numonly(o,17)/districtclients2(2,17);
            o=o+1;

        elseif                  saididailytemp(i,2)==45                 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0
```

```
saididaily_district_numonly(p,18)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytem
p(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(p,18)=saididaily_district_numonly(p,18)/districtclients2(2,18);
        p=p+1;

    elseif saididailytemp(i,2)==50 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(q,20)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytem
p(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(q,20)=saididaily_district_numonly(q,20)/districtclients2(2,20);
        q=q+1;

    elseif saididailytemp(i,2)==53 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(s,21)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytem
p(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(s,21)=saididaily_district_numonly(s,21)/districtclients2(2,21);
        s=s+1;

    elseif saididailytemp(i,2)==54 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(t,22)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytem
p(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(t,22)=saididaily_district_numonly(t,22)/districtclients2(2,22);
        t=t+1;

    elseif saididailytemp(i,2)==55 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0
```

```
saididaily_district_numonly(v,23)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(v,23)=saididaily_district_numonly(v,23)/districtclients2(2,23);
        v=v+1;

    elseif saididailytemp(i,2)==60 && (saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(w,25)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(w,25)=saididaily_district_numonly(w,25)/districtclients2(2,25);
        w=w+1;

    elseif saididailytemp(i,2)==61 && (saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(x,26)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(x,26)=saididaily_district_numonly(x,26)/districtclients2(2,26);
        x=x+1;

    elseif saididailytemp(i,2)==62 && (saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(y,27)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(y,27)=saididaily_district_numonly(y,27)/districtclients2(2,27);
        y=y+1;

    elseif saididailytemp(i,2)==63 && (saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0
```

```
saididaily_district_numonly(z,28)=saididailytemp(i,4)+saididailytemp(i,5)+saididailytem
p(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10);

saididaily_district_div(z,28)=saididaily_district_numonly(z,28)/districtclients2(2,28);
            z=z+1;

        elseif                  saididailytemp(i,2)==70                 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(aa,30)=saididailytemp(i,4)+saididailytemp(i,5)+saididailyte
mp(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10
);

saididaily_district_div(aa,30)=saididaily_district_numonly(aa,30)/districtclients2(2,30);
            aa=aa+1;

        elseif                  saididailytemp(i,2)==71                 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(bb,31)=saididailytemp(i,4)+saididailytemp(i,5)+saididailyte
mp(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10
);

saididaily_district_div(bb,31)=saididaily_district_numonly(bb,31)/districtclients2(2,31);
            bb=bb+1;

        elseif                  saididailytemp(i,2)==72                 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(cc,32)=saididailytemp(i,4)+saididailytemp(i,5)+saididailyte
mp(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10
);

saididaily_district_div(cc,32)=saididaily_district_numonly(cc,32)/districtclients2(2,32);
            cc=cc+1;
```

```
        elseif saididailytemp(i,2)==73 &&
(saididailytemp(i,4)+saididailytemp(i,5)+saididailytemp(i,6)+saididailytemp(i,7)+saidida
ilytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10))>0


saididaily_district_numonly(dd,33)=saididailytemp(i,4)+saididailytemp(i,5)+saididailyte
mp(i,6)+saididailytemp(i,7)+saididailytemp(i,8)+saididailytemp(i,9)+saididailytemp(i,10
);

saididaily_district_div(dd,33)=saididaily_district_numonly(dd,33)/districtclients2(2,33);
            dd=dd+1;
        end

      end
   end

   header3=[a,b,c,d,e,0,f,g,h,0,j,k,l,0,m,n,o,p,0,q,s,t,v,0,w,x,y,z,0,aa,bb,cc,dd];

   for dist_col=1:33

      if (dist_col~=6 && dist_col~=10 && dist_col~=14 && dist_col~=19 &&
dist_col~=24 && dist_col~=29)
        dist_rows=header3(1,dist_col)-1;
        saididailytemp=zeros((dist_rows-3),1);
        saididailytemp2=zeros((dist_rows-3),1);

        for index=3:dist_rows

           saididailytemp(index-2,1)=saididaily_district_numonly(index,dist_col);
           saididailytemp2(index-2,1)=saididaily_district_div(index,dist_col);
        end

        %saididailytempdiv=saididailytemp/districtclients2(2,dist_col);

        if dist_col==22

           test=saididailytemp2;
           test2=saididailytemp;

        end

        theta=mean(log(saididailytemp2)); %theta = mean in Normal mode
        dev=std(log(saididailytemp2));    %omega = variance in Normal mode
        omega=dev^2;
        EX=exp((theta+omega/2));
```

```
   VX=exp((2*theta+omega))*(exp(omega)-1);

   EXVX_district(3,dist_col)=EX;
   EXVX_district(4,dist_col)=VX;

end
if dist_col<=32
   if header1(1,dist_col)==header1(1,dist_col+1) || header1(1,dist_col+1)==99

      saididailytemplong=vertcat(saididailytemplong,saididailytemp);



   end
else

   saididailytemplong=vertcat(saididailytemplong,saididailytemp);
end

if dist_col==5

   saididaily_region_0=saididailytemplong;
   saididaily_region_0(1,:)=[];
   saididailytemplong=0;


end

if dist_col==10

   saididaily_region_1=saididailytemplong;
   saididaily_region_1(1,:)=[];
   saididailytemplong=0;


end

if dist_col==14

   saididaily_region_2=saididailytemplong;
   saididaily_region_2(1,:)=[];
   saididailytemplong=0;


end
```

```
    if dist_col==19

        saididaily_region_3=saididailytemplong;
        saididaily_region_3(1,:)=[];
        saididailytemplong=0;


    end

    if dist_col==24

        saididaily_region_4=saididailytemplong;
        saididaily_region_4(1,:)=[];
        saididailytemplong=0;


    end

    if dist_col==29

        saididaily_region_5=saididailytemplong;
        saididaily_region_5(1,:)=[];
        saididailytemplong=0;


    end

    if dist_col==33

        saididaily_region_6=saididailytemplong;
        saididaily_region_6(1,:)=[];
        saididailytemplong=0;


    end

end

for r=0:6

    if r==0
        saididailytemp=saididaily_region_0;

    end
```

```
    if r==1
       saididailytemp=saididaily_region_1;


    end
    if r==2
       saididailytemp=saididaily_region_2;


    end
    if r==3
       saididailytemp=saididaily_region_3;


    end
    if r==4
       saididailytemp=saididaily_region_4;


    end
    if r==5
       saididailytemp=saididaily_region_5;


    end
    if r==6
       saididailytemp=saididaily_region_6;


    end

    saididailytemp=saididailytemp/regionclients(r+1,2);

    theta=mean(log(saididailytemp)); %theta = mean in Normal mode
    dev=std(log(saididailytemp));    %omega = variance in Normal mode
    omega=dev^2;
    EX=exp((theta+omega/2));
    VX=exp((2*theta+omega))*(exp(omega)-1);

    EXVX_region(2,r+1)=EX;
    EXVX_region(3,r+1)=VX;

end

saididaily_region_0=saididaily_region_0/regionclients(1,2);
```

```
saididaily_region_1=saididaily_region_1/regionclients(2,2);
saididaily_region_2=saididaily_region_2/regionclients(3,2);
saididaily_region_3=saididaily_region_3/regionclients(4,2);
saididaily_region_4=saididaily_region_4/regionclients(5,2);
saididaily_region_5=saididaily_region_5/regionclients(6,2);
saididaily_region_6=saididaily_region_6/regionclients(7,2);

skewmat(1,1)=skewness(saididaily_region_0,1);
skewmat(2,1)=skewness(saididaily_region_1,1);
skewmat(3,1)=skewness(saididaily_region_2,1);
skewmat(4,1)=skewness(saididaily_region_3,1);
skewmat(5,1)=skewness(saididaily_region_4,1);
skewmat(6,1)=skewness(saididaily_region_5,1);
skewmat(7,1)=skewness(saididaily_region_6,1);

if select==1

   EXVX_Uave(1,1)=mean(EXVX_district(3,:));
   EXVX_Uave(2,1)=mean(EXVX_district(4,:));
   save saididaily_breaker.mat EXVX_district EXVX_region EXVX_Uave
   save saididaily_breaker.mat saididaily_* -append
   save saididaily_breaker.mat districtclients2 regionclients -append
   save saididaily_breaker.mat skewmat -append
   save saididaily_breaker_div.txt saididaily_district_div -ascii
   save saididaily_breaker_num.txt saididaily_district_numonly -ascii


elseif select==2

   if option==1
      save saididaily_breaker_adj_op1.mat EXVX_district EXVX_region
      save saididaily_breaker_adj_op1.mat saididaily_* -append
      save saididaily_breaker_adj_op1.mat districtclients2 regionclients -append
      save saididaily_breaker_adj_op1.mat skewmat -append


   elseif option==2
      save saididaily_breaker_adj_op2.mat EXVX_district EXVX_region
      save saididaily_breaker_adj_op2.mat saididaily_* -append
      save saididaily_breaker_adj_op2.mat districtclients2 regionclients -append
      save saididaily_breaker_adj_op2.mat skewmat -append


   elseif option==3
      save saididaily_breaker_adj_op3.mat EXVX_district EXVX_region
```

```
        save saididaily_breaker_adj_op3.mat saididaily_* -append
        save saididaily_breaker_adj_op3.mat districtclients2 regionclients -append
        save saididaily_breaker_adj_op3.mat skewmat -append


    elseif option==4
        save saididaily_breaker_adj_op4.mat EXVX_district EXVX_region
        save saididaily_breaker_adj_op4.mat saididaily_* -append
        save saididaily_breaker_adj_op4.mat districtclients2 regionclients -append
        save saididaily_breaker_adj_op4.mat skewmat -append


    end

  end

%end

toc
```

## Pre-WEIGHTER Script:

```
%This new script creates the weight matrix using the ideas set forth in
%log7705.
clear

load districtFAULTS.mat
load saididaily_breaker.mat

districtFAULT2=transpose(districtFAULT);
%districtWEIGHT=zeros(27,33);
districtWEIGHT2=zeros(32,33);
districtWEIGHT2(1:2,:)=saididaily_district_numonly(1:2,:);

%districtWEIGHT(:,1:3)=districtFAULT(:,1:3);
districtFAULT_total=zeros(1,30);
temp=sum(districtFAULT(:,4:34));
districtFAULT_total(1,3:33)=temp;
districtFAULT_PU(1,3:33)=districtFAULT_total(:,3:33)/sum(districtFAULT_total(:,3:33));

districtFAULT_PURANK=ones(3,33);
%sorty=zeros(1,33);

%[sorty,districtFAULT_PURANK(1,3:33)]=sort(districtFAULT_total(1,3:33),'ascend'); %This provides
the ranking. THe sort function provides the index for the sorted values, which are discarded.
[sorty,districtFAULT_PURANK(1,3:33)]=sort(districtFAULT_PU(1,3:33),'ascend');


districtFAULT_PURANK(2,3:33)=32-districtFAULT_PURANK(1,3:33); %Row 1=rank it was Row 2=
rank it now is Row3=PU value paired
```

```
for i=3:33

   districtFAULT_PURANK(3,i)=sorty(1,districtFAULT_PURANK(2,i)); %This row 3 contains the PU value paired to the original PU value. Bigger with smaller.
end

j=1;
for i=1:33

   if districtWEIGHT2(1,i)~=99

      districtWEIGHT2(3:33,i)=districtFAULT2(4:34,j); %Valores Originales
      districtWEIGHT2b(3:33,i)=districtFAULT2(4:34,j)/sum(districtFAULT2(4:34,j)); %Valores PU
      j=j+1;

   end

end

districtFAULT_PURANK2=transpose(districtFAULT_PURANK); %Change to vertical to match districtWEIGHTS2

for i=3:33

   districtFAULT_mean(i,1)=mean(districtWEIGHT2(i,:));%Mean value per fault type. Vertical conf

end


for i=3:33

   districtFAULT_meanPU(i,1)=districtFAULT_mean(i,1)/sum(districtFAULT_mean(:,1)); %Mean value in PU per fault type. Vertical conf

end

for i=3:33

   districtFAULT_max(i,1)=max(districtWEIGHT2(i,:));%Max value per fault type. Vertical conf

end


for i=3:33

   districtFAULT_maxPU(i,1)=districtFAULT_max(i,1)/sum(districtFAULT_max(:,1)); %Mean value in PU per fault type. Vertical conf

end


districtWEIGHT_paired=zeros(size(districtWEIGHT2));
```

```
districtWEIGHT_meaned=zeros(size(districtWEIGHT2));
districtWEIGHT_maxed=zeros(size(districtWEIGHT2));
districtWEIGHT_paired(1:2,:)=saididaily_district_numonly(1:2,:);
districtWEIGHT_meaned(1:2,:)=saididaily_district_numonly(1:2,:);
districtWEIGHT_maxed(1:2,:)=saididaily_district_numonly(1:2,:);


for i=1:33

   if districtWEIGHT2(1,i)~=99
      for j=3:33
         if districtFAULT_PURANK2(j,3)>=districtWEIGHT2b(j,i)
            districtWEIGHT_paired(j,i)=districtWEIGHT2(j,i)*(1+districtFAULT_PURANK2(j,3));
%Originales Ajustados no PU
         elseif districtFAULT_PURANK2(j,3)<districtWEIGHT2b(j,i)
            districtWEIGHT_paired(j,i)=districtWEIGHT2(j,i)*(1-districtFAULT_PURANK2(j,3));
         end
      end

   end
end

for i=1:33

   if districtWEIGHT2(1,i)~=99
      for j=3:33
         if districtFAULT_meanPU(j,1)>=districtWEIGHT2b(j,i)
            districtWEIGHT_meaned(j,i)=districtWEIGHT2(j,i)*(1+districtFAULT_meanPU(j,1));
%Originales Ajustados no PU
         elseif districtFAULT_meanPU(j,1)<districtWEIGHT2b(j,i)
            districtWEIGHT_meaned(j,i)=districtWEIGHT2(j,i)*(1-districtFAULT_meanPU(j,1));
         end
      end

   end
end

for i=1:33

   if districtWEIGHT2(1,i)~=99
      for j=3:33
         if districtFAULT_maxPU(j,1)>=districtWEIGHT2b(j,i)
            districtWEIGHT_maxed(j,i)=districtWEIGHT2(j,i)*(1+districtFAULT_maxPU(j,1));
%Originales Ajustados no PU
         elseif districtFAULT_maxPU(j,1)<districtWEIGHT2b(j,i)
            districtWEIGHT_maxed(j,i)=districtWEIGHT2(j,i)*(1-districtFAULT_maxPU(j,1));
         end
      end

   end
end

%----Normalizing section----------------------------
for i=1:33
```

```
   if districtWEIGHT2(1,i)~=99

districtWEIGHT_pairedPU(3:33,i)=districtWEIGHT_paired(3:33,i)/sum(districtWEIGHT_paired(3:33,i));
%Originales Ajustados PU

   end

end

for i=1:33

   if districtWEIGHT2(1,i)~=99

districtWEIGHT_meanedPU(3:33,i)=districtWEIGHT_meaned(3:33,i)/sum(districtWEIGHT_meaned(3:33
,i)); %Originales Ajustados PU

   end

end

for i=1:33

   if districtWEIGHT2(1,i)~=99

districtWEIGHT_maxedPU(3:33,i)=districtWEIGHT_maxed(3:33,i)/sum(districtWEIGHT_maxed(3:33,i))
; %Originales Ajustados PU

   end

end
save weights.mat *WEIGHT*
```

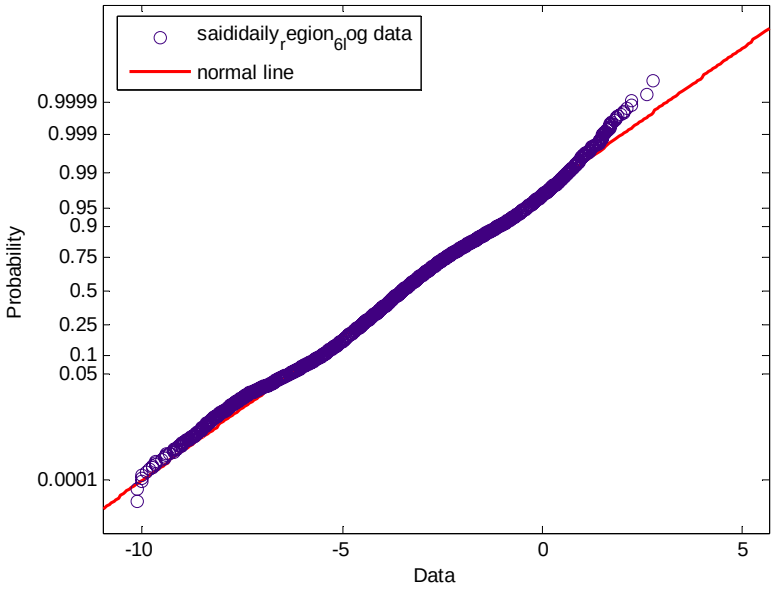
## WEIGHTER Function:

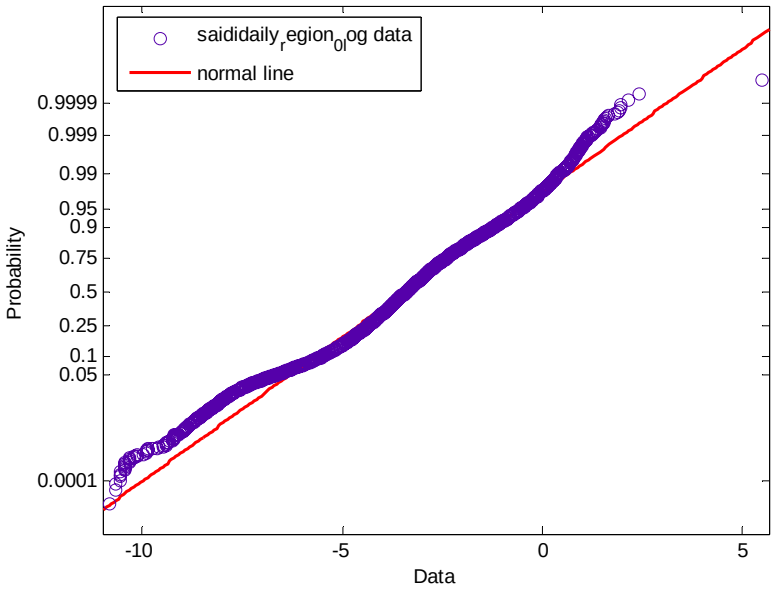
```
function [saidiADJUSTED]=saidiADJUSTER(saidiDAILY,district,faultcode,option)

districtheader=[20,22,23,24,95,0,10,11,12,0,13,14,34,0,40,42,44,45,0,50,53,54,55,0,60,61,62,63,0,70,71,72
,73];

load faultcolumn.mat
load weights.mat
load saididaily_breaker.mat EXVX_district EXVX_region EXVX_Uave

%Must change options and eliminate calculations.Options will only make
%multiplications here. All weighting is done in WEIGHTMAKER. New 3 options.
%Option 1: Using the paired method. HIgher with smaller, then PU using SUM.
%Option 2: PU using MEAN values for each FAULT.
%Option 3: Using ACTUAL/MAX vaue of all districts.
```

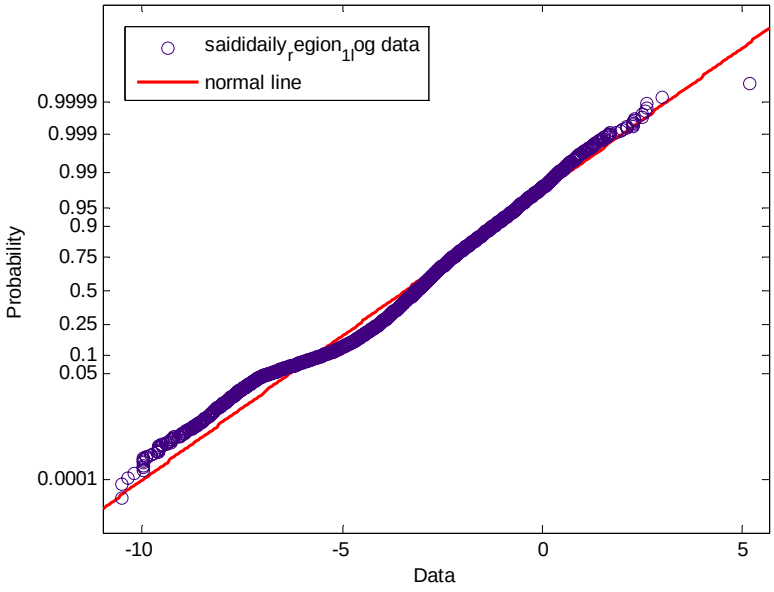
```
if option==1
   for i=1:33 %Move along the columns of the weight square

      for j=1:31 %Move along the rows of the weight square

         if districtheader(1,i)==district && faultcolumn(j,1)==faultcode

            saidiADJUSTED=saidiDAILY*(districtWEIGHT_pairedPU(j+2,i)); %WEIGHTs loaded from
weights.mat
            break

         elseif
faultcode~=[13,38,39,48,51,52,53,54,56,58,59,63,65,66,67,69,83,85,86,87,88,89,90,91,92,93,94,95,96,97,9
8]

            saidiADJUSTED=saidiDAILY;
            break
         end

      end

   end

elseif option==2 %same as 1 but add instead of substract the weight

   for i=1:33 %Move along the columns of the weight square

      for j=1:31 %Move along the rows of the weight square

         if districtheader(1,i)==district && faultcolumn(j,1)==faultcode

            saidiADJUSTED=saidiDAILY*(districtWEIGHT_meanedPU(j+2,i)); %WEIGHTs loaded from
weights.mat
            break

         elseif
faultcode~=[13,38,39,48,51,52,53,54,56,58,59,63,65,66,67,69,83,85,86,87,88,89,90,91,92,93,94,95,96,97,9
8]

            saidiADJUSTED=saidiDAILY;
            break
         end

      end

   end

elseif option==3

   for i=1:33 %Move along the columns of the weight square

      for j=1:31 %Move along the rows of the weight square
```

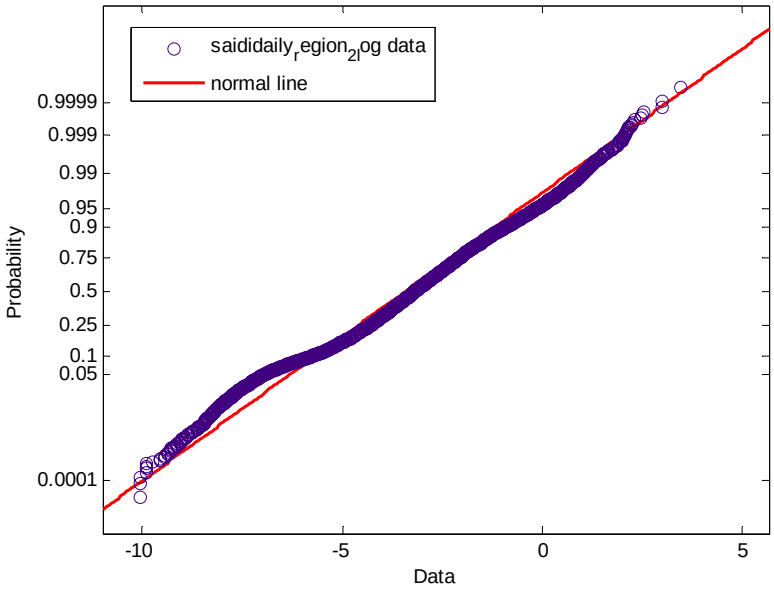
```
        if districtheader(1,i)==district && faultcolumn(j,1)==faultcode

            saidiADJUSTED=saidiDAILY*(districtWEIGHT_maxedPU(j+2,i)); %WEIGHTs loaded from
weights.mat
            break

        elseif
faultcode~=[13,38,39,48,51,52,53,54,56,58,59,63,65,66,67,69,83,85,86,87,88,89,90,91,92,93,94,95,96,97,9
8]
            saidiADJUSTED=saidiDAILY;
            break
        end
      end
   end
end
```

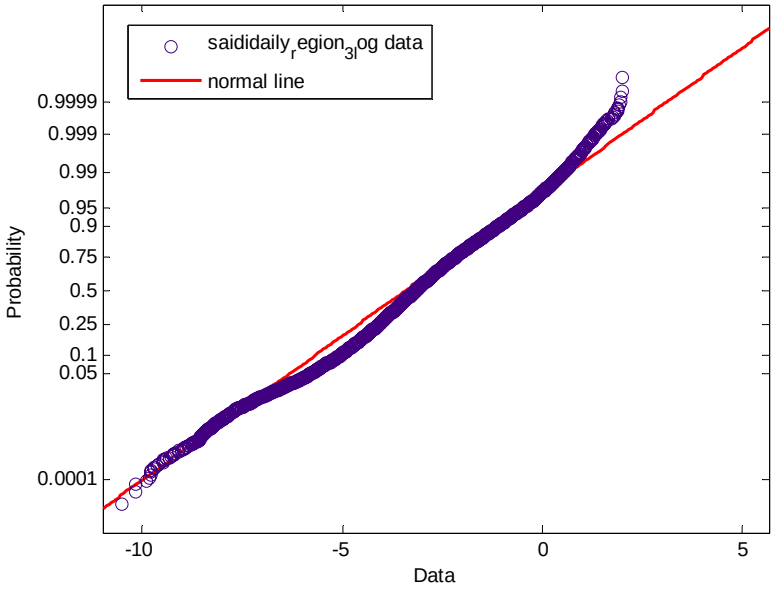
# APPENDIX D: STATISTICAL FITS FOR ALL REGIONS

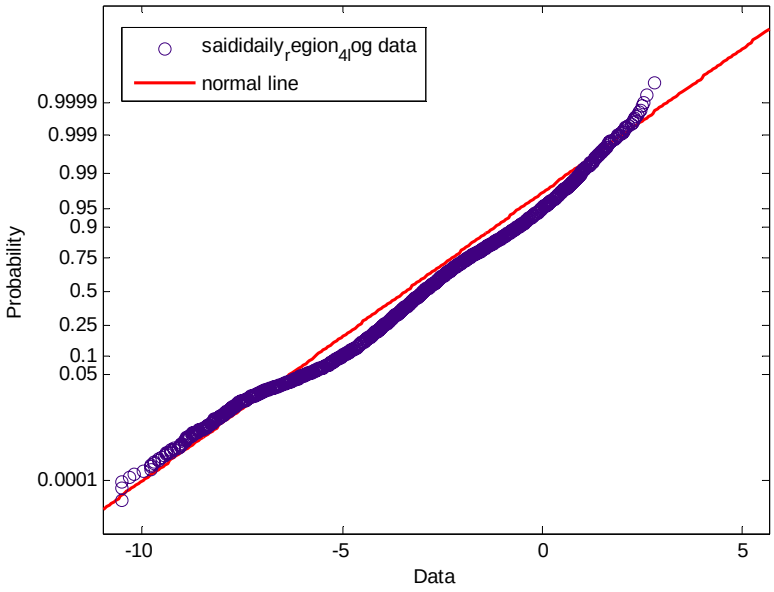


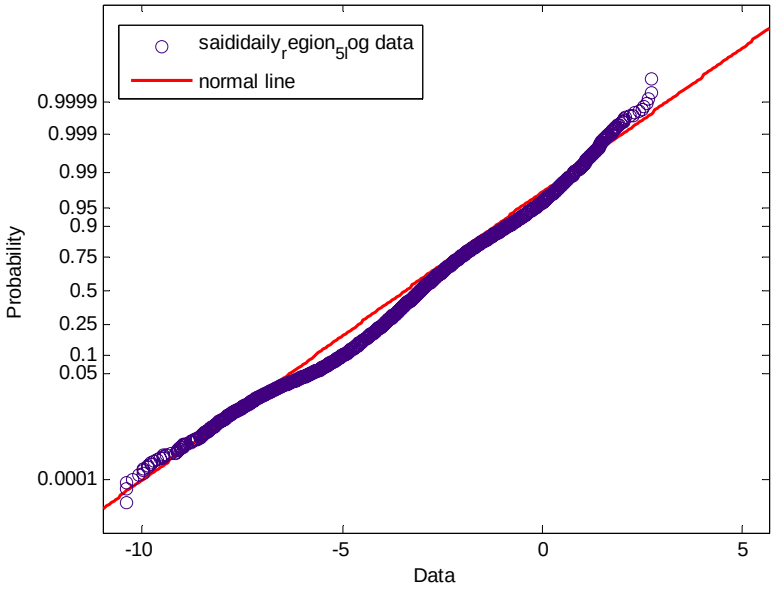




saididaily_region_1log data
normal line
Probability
0.9999
0.999
0.99
0.95
0.9
0.75
0.5
0.25
0.1
0.05
0.0001
-10
-5
0
5
Data


saididaily_region_2log data
normal line
Probability
0.9999
0.999
0.99
0.95
0.9
0.75
0.5
0.25
0.1
0.05
0.0001
-10
-5
0
5
Data


saididaily_region_3log data
normal line
Probability
0.9999
0.999
0.99
0.95
0.9
0.75
0.5
0.25
0.1
0.05
0.0001
-10
-5
0
5
Data


saididaily_region_4log data
normal line
Probability
0.9999
0.999
0.99
0.95
0.9
0.75
0.5
0.25
0.1
0.05
0.0001
-10
-5
0
5
Data


saididaily_region_5log data
normal line
0.9999
0.999
0.99
0.95
0.9
0.75
0.5
0.25
0.1
0.05
0.0001
Probability
-10
-5
0
5
Data