MnROAD Data Mining, Evaluation and Quantification Phase I

Minnesota Department of Transportation

RESEARCH SERVICES

Office of Policy Analysis, Research & Innovation

Randal J. Barnes, Principal Investigator
Department of Civil Engineering
University of Minnesota

July 2010

Research Project
Final Report 2010-26

LRRB
LOCAL ROAD RESEARCH BOARD

Your Destination...Our Priority

Technical Report Documentation Page

1. Report No.	2.	3. Recipients Accession No.
MN/RC 2010-26		

4. Title and Subtitle	5. Report Date
MnROAD Data Mining, Evaluation and Quantification – Phase I	July 2010
	6.
7. Author(s) Barnes, Randal J.	8. Performing Organization Report No.
9. Performing Organization Name and Address University of Minnesota Department of Civil Engineering 500 Pillsbury Drive SE Minneapolis, MN 55455	10. Project/Task/Work Unit No. CTS Project #2008046 11. Contract (C) or Grant (G) No. (c) 89261 (wo) 52
12. Sponsoring Organization Name and Address Minnesota Department of Transportation Research Services Section 395 John Ireland Boulevard, MS 330 St. Paul, MN 55155-1899	13. Type of Report and Period Covered Final Report 14. Sponsoring Agency Code

15. Supplementary Notes
http://www.lrrb.org/pdf/201026.pdf

16. Abstract (Limit: 250 words)

A data filtering system for the MnROAD temperature database was designed and implemented. Fourteen inter-dependent quantitative tests were developed to identify and flag erroneous, questionable, or exceptional data. Four of the tests identify missing and intermittent data streams. Three of the tests analyze the time series from individual sensors and identify outliers. Three of the tests compare data streams of similar sensors; “similar” implies identical pavement type, general location, and sensor depth. The remaining four tests are summary tests that identify periods of unreliable data.

The specific analysis and quantitative results are based upon the 471,178,324 data records from 1,313 thermocouple sensors in 48 MnROAD test cells collected from 1 January 1996 through October 2007. The considered test cells include both hot mix asphalt and Portland cement concrete sections from both the Mainline and Low Volume Road.

The majority of the sensors performed very well: 714 of the 1,282 operational sensors produced reliable data more than 99 percent of the time. Only 18 of 1,282 operational sensors produce reliable data less than 50 percent of the time. Only 31 of the original 1,313 sensor were wholly non-operational.

A wide variety of statistical tables and graphical representation were produced in a digital format for the considered data.

Although this project focuses on a particular set of data, the concepts and tools developed in this project are designed to be extensible to accommodate the filtering of the ongoing and future data collection efforts at MnROAD.

17. Document Analysis/Descriptors	18. Availability Statement
MnROAD, Thermocouples, Pavements, Pavement temperature, Data processing operations, Data filtering, Data mining	No restrictions. Document available from: National Technical Information Services, Springfield, Virginia 22161

19. Security Class (this report)	20. Security Class (this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	250	

MnROAD Data Mining, Evaluation and Quantification
Phase I

Final Report

Prepared by:

Randal J. Barnes

Department of Civil Engineering
University of Minnesota

July 2010

Published by:

Minnesota Department of Transportation
Research Services Section
395 John Ireland Boulevard, MS 330
St. Paul, Minnesota 55155-1899

Acknowledgments

The author of this report acknowledges the consistently constructive criticism from, and informative dialog with, the members of the project's Technical Advisory Panel. The efforts of two members of the panel demand specific acknowledgment: Mr. Tom Burnham and Mr. Dennis Carroll.

Contents

List of Figures

List of Tables

Executive Summary

A data filtering system for the MnROAD temperature database was designed and implemented. Fourteen inter-dependent quantitative tests were developed to identify and flag erroneous, questionable, or exceptional data. Four of the tests identify missing and intermittent data streams. Three of the tests analyze the time series from individual sensors and identify outliers. Three of the tests compare data streams of similar sensors; "similar" implies identical pavement type, general location, and sensor depth. The remaining four tests are summary tests that identify periods of unreliable data.

The specific analysis and quantitative results are based upon the 471,178,324 data records from 1,313 thermocouple sensors in 48 MnROAD test cells collected from 1 January 1996 through October 2007. The considered test cells include both hot mix asphalt and Portland cement concrete sections from both the interstate and low-volume roadways.

The majority of the sensors performed very well: 714 of the 1,282 operational sensors produced reliable data more than 99 percent of the time. Only 18 of 1,282 operational sensors produce reliable data less than 50 percent of the time. Only 31 of the original 1,313 sensor were wholly non-operational.

A wide variety of statistical tables and graphical representation were produced in a digital format for the considered data.

Although this project focuses on a particular set of data, the concepts and tools developed in this project are designed to be extensible to accommodate the filtering of the ongoing and future data collection efforts at Mn/ROAD.

Chapter 1

Introduction

1.1 Purpose of this Report

The database at MnROAD is one of the main products of MnROAD 's first decade of operation. It contains valuable information for in-depth pavement research studies for cold climates. Its size and impact will be compared with the premier database in the United States for pavement performance data, the Long-Term Pavement Performance (LTPP) database. At the same time, only small portions of the data contained in the very large MnROAD database have been extracted and analyzed for various pavement related research studies. The presence of some erroneous records and the small number of tables with the processed data have limited the use of the database.

This report details the design and implementation of a data filtering system for the MnROAD thermocouple temperature database. The specific analysis and quantitative results presented in this report are based upon the $471,178,324$ data records from $1,313$ thermocouple sensors in 48 pavement test cells collected from 1 January 1996 through October 2007. The cells include both hot mix asphalt (HMA) and Portland cement concrete (PCC) sections from both the interstate and low-volume roadways.

Although this report focuses on a particular set of data, the concepts and tools developed in this project are extensible to accommodate the filtering of future data collection efforts.

1.2 MnROAD

The MnROAD facility consists of two roads segments lying parallel to Interstate 94 outside Otsego, Minnesota (40 miles northwest of Minneapolis/Saint Paul, Minnesota). The site is comprised of a 3.5-mile "mainline" roadway carrying live interstate traffic, plus a 2.5-mile low-volume loop road where controlled truck weight and traffic volume simulate conditions on rural roads. MnROAD 's 50+ test cells, each up to 500 feet in length, are paved with different thickness of concrete, asphalt, and aggregate. The cells are distributed over the two roadways to represent a wide range of pavement types, with varying combinations of surface, base, subbase, subgrade, drainage, and compaction. For a more detailed background on MnROAD see [16].

The MnROAD facility was conceived in 1987 and constructed in 1994. The data collected from the MnROAD test cells (40 of which collect temperature data) has been used by researchers around the world. For more information on the history of MnROAD see [19].

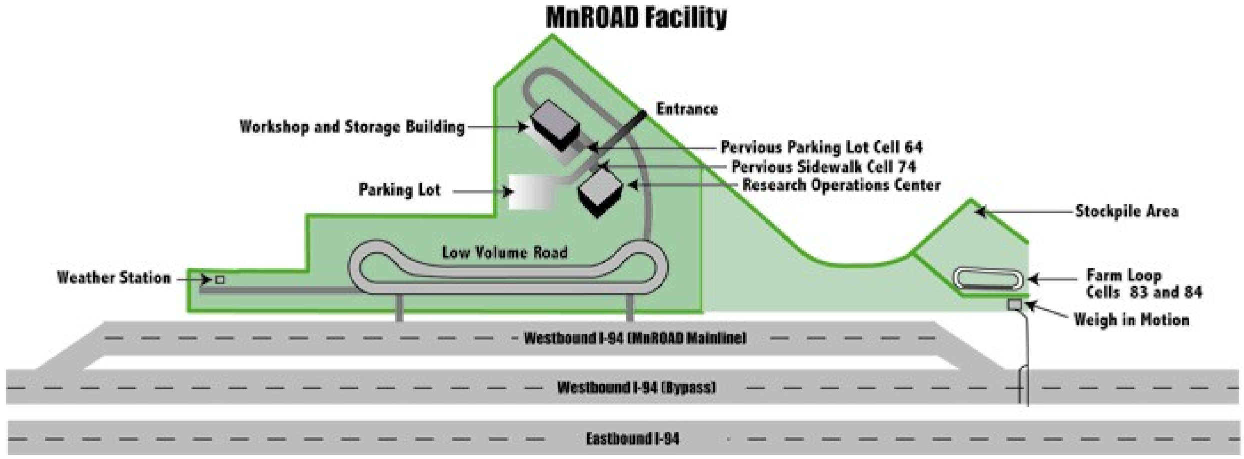

Figure 1.1: The MnROAD facility layout shows the Mainline and the the Low Volume Road.

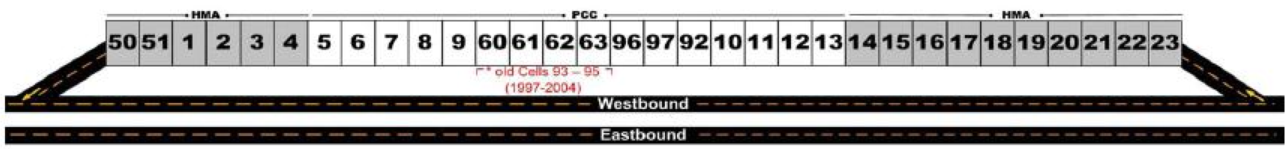

Figure 1.2: The cell layout, and numbering, is shown for the Mainline sections.

Due to the mid-continent climatic conditions of central Minnesota, MnROAD is situated to experience some of the hardest winters in the continental United States. This environment, paired with the traffic flow of Interstate 94, prompted the Minnesota Department of Transportation (Mn/DOT) and the Minnesota Local Road Research Board (LRRB) to shoulder the majority of the 25 million dollar cost[1] of this one-of-a-kind facility. Continued funding for MnROAD comes from a combination of industry, the Local Road Research Board, federal and pooled state funds. For more information on the continued funding of the MnROAD project see [18].

The MnROAD facility has two separate roadways: the "Mainline" and the "Low Volume Road" (see Figure 1.1, which was extracted from [17]). The larger of the two MnROAD roadways, known as the "Mainline" (see Figure 1.2), allows researchers to accurately simulate real road conditions by rerouting traffic from the westbound lanes of Interstate 94 onto the 3.5-mile test road in the desired frequency and volume. Furthermore, all traffic can be easily routed off of the test sections to allow for researcher interaction. The traffic on the 2.5-mile "Low Volume Road" (see Figure 1.3) is simulated using a facility run 5-axle semi-tractor-trailer that repeatedly drives the loop.

1.3 Objective of this Project

The objective of this project is to improve effectiveness and quality of the MnROAD pavement temperature record database by two means:

1. identify, flag, and eliminate questionable, erroneous, and unreliable data; and

[1]This estimated figure of 25 million dollars includes construction and the first 10 years of operation.

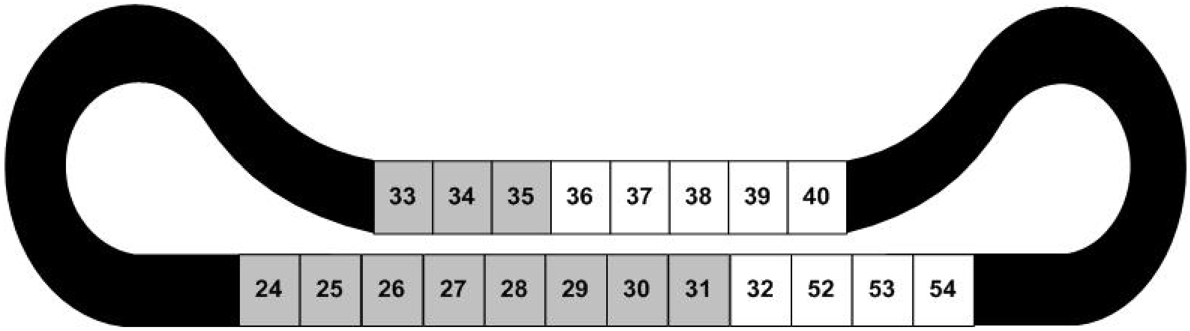

Figure 1.3: The cell layout, and numbering, is shown for the Low Volume Road sections.

2. create summary tables from the reliable data.

The computed parameters consist of statistical summaries of the raw measured data or values computed from the raw data using the appropriate models. This will enable a wider use of the MnROAD data and increase the overall impact of MnROAD project.

1.4 Report Organization

The considered, extensive, data set, and some of the specific issues faced when dealing with this data, are presented in Chapter 2. Chapter 3 presents a collection of examples of erroneous data behavior found in the considered data set. These examples are meant to be demonstrative and not exhaustive, they are included as a reference tool for the subsequent presentation. Chapter 4 introduces the tests, checks, and flags that are detailed in Chapter 6. Chapter 5 presents the necessary statistical background required by the test, checks, and flags that are detailed in Chapter 6. Chapter 6 presents an exhaustive enumeration of the test, checks, and flags designed and implemented in this project. Chapter 7 presents some unique details of the analysis of temporal variations. Chapter 8 is a lengthy compilation of results, and the associated summary. The details of the software are presented in the appendices.

Chapter 2

Data Set

2.1 Background

The guide to the MnROAD Database [45], and internal reports by various researchers [8, 4, 6, 12, 11] provide an excellent overview of MnROAD 's experience in data acquisition and verification.

All of the data generated at the MnROAD facility is free and open to the public. Requests for customized data reports can be made via the web page: `http://www.dot.state.mn.us/mnroad/data/requestspecial.html`.

2.2 General Description

In initiating this project, the Mn/DOT professional staff extracted the following data from the MnROAD database for validation and preliminary trend analysis. The complete data set considered includes:

- 683 raw data files (.csv)
- Approximately 4.4 Gigabytes of raw data
- 48 Test cells (1-40, 52-54, 60, 62, 93, 95, and 97)
- 148 sensor groups (2-12 sensors per group)
- 1305 individual sensors (TC data)
- 327 sensors that are inactive (retired or dead)
- Time span for considered data: 1996 to 2007
- Measurements every 15 minutes

Data considered in this project spans 1 January 1996 to 3 October 2007. However, the data from individual cells begin and end at a variety of times. Figure 2.1 gives a graphical representation of each cell's period of activity.

A summary of all considered data is given in Tables 2.1, 2.2, and 2.3.

Table 2.1: A tabular summary of all considered MnROAD temperature sensor data.

Cell	# Tree	# Sens	Begin	End	# Dead[a]	Notes
1	1	11	1996	2007	0	Seq # 4-11 n/a prior to 1/1/1998.
2	1	11	1996	2007	0	Seq # 4-11 n/a prior to 1/1/1998.
3	1	11	1996	2007	0	Seq # 5-11 n/a prior to 1/1/1998.
4	7	69	1996	2007	9	Seq # 5-10, 15-20, 26-30, 35-39, 44-49, 55-59, and 64-69 n/a prior to 1/1/1998.
5	1	12	1996	2007	1	Seq # 6-11 n/a prior to 1/1/1998.
6	7	75	1996	2007	3	Seq # 5-11, 17-22, 27-33, 36-42, 47-53, 58-64, and 68-75 n/a prior to 1/1/1998.
7	7	75	1996	2007	2	Seq # 5-11, 16-22, 27-33, 36-42, 46-53, 58-64, and 69-75 n/a prior to 1/1/1998.
8	1	12	1996	2007	1	Seq # 5-12 n/a prior to 1/1/1998.
9	1	12	1996	2007	0	Seq # 5-12 n/a prior to 1/1/1998.
10	4	42	1996	2007	0	Seq # 6-11, 17-22, 28-33, and 36-42 n/a prior to 1/1/1998.
11	1	11	1996	2007	2	Seq # 6-11 n/a prior to 1/1/1998; Sensors 10-11 failed late 2004.
12	4	42	1996	2007	6	Seq # 6-11, 17-22, 28-33, and 36-41 n/a prior to 1/1/1998;
						Seq # 42 never came on line.
13	1	11	1996	2007	0	Seq # 6-11 n/a prior to 1/1/1998.
14	1	10	1996	2007	0	Seq # 6-10 n/a prior to 1/1/1998.
15	4	37	1996	2007	0	Seq # 5-10, 15-20, and 26-37 n/a prior to 1/1/1998.
16	1	11	1996	2006	0	Seq # 5-11 n/a prior to 1/1/1998.
17	7	76	1996	2006	8	Seq # 5-11, 16-22, 27-33, 37-43, 48-57, 59-65, and 70-76 n/a prior to 1/1/1998.
18	1	11	1996	2006	4	Seq # 1, 3, 5, and 7-11 n/a prior to 1/1/1998.
19	1	11	1996	2006	0	Seq # 6-11 n/a prior to 1/1/1998.
20	1	11	1996	2006	0	Seq # 5-11 n/a prior to 1/1/1998.
21	1	11	1996	2006	0	Seq # 5-11 n/a prior to 1/1/1998.

[a] As of 2007

Table 2.2: A tabular summary of all considered MnROAD temperature sensor data (continued).

Cell	# Tree	# Sens	Begin	End	# Dead[a]	Notes
22	7	76	1996	2006	0	Seq # 5-11, 16-22, 27-33, 37-43, 48-54, 59-65, and 70-76 n/a prior to 1/1/1998.
23	1	11	1996	2007	0	Seq # 6-10 n/a prior to 1/1/1998.
24	1	10	1996	2006	0	Seq # 4-10 n/a prior to 1/1/1998.
25	4	37	1996	2006	0	Seq # 4-10, 14-20, and 25-37 n/a prior to 1/1/1998.
26	4	37	1996	2001	37	Seq # 1-37 removed 8/8/2001.
	1	10	2004	2007	1	Seq # 101-110 added 11/29/2004.
27	4	37	1996	1999	37	Seq # 1-37 removed 7/21/1999.
	7	28	2000	2006	2	Seq # 101-128 added 3/8/2000.
28	1	10	1996	1999	10	Seq # 1-10 removed 7/22/1999
	6	28	2000	2006	28	Seq # 101-128 added 3/11/2000, then removed 6/9/2006.
	1	4	2007	2007	2	Seq # 201-204 added 3/6/2007.
29	1	11	1996	2007	0	Seq # 4-11 not available prior to 1/1/1998.
30	4	37	1996	2007	5	Seq # 3-10, 14-20, and 25-37 n/a prior to 1/1/1998;
						Seq # 1-5 removed 7/6/2007.
31	1	10	1996	2007	2	Seq # 2 and 10 went dead summer 2001.
	1	16	2004	2007	0	Seq # 101-116 added 12/29/2004.
32	1	9	1998	2007	0	Seq # 1-9 were n/a prior to 1/1/1998.
	2	16	2000	2007	6	Seq # 101-116 added 6/30/2000, removed 9/21/2006.
33	4	34	1998	1999	34	Seq # 1-34 removed 7/21/1999.
	3	6	2000	2007	0	Seq # 101-106 added 3/8/2000.
34	1	9	1998	1999	9	Seq # 1-9 removed 7/21/1999.
	3	17	2000	2007	13	Seq # 101-117 added 3/8/2000; Sensors 103-109 dead Spring 2002.

[a] As of 2007

Table 2.3: A tabular summary of all considered MnROAD temperature sensor data (continued).

Cell	# Tree	# Sens	Begin	End	# Dead[a]	Notes
35	7	61	1998	2000	61	Seq # 1-16 removed 1/05/2000; Sensors 17-61 removed 10/26/1999.
	3	7	2000	2007	0	Seq # 101-107 added 3/15/2000.
36	1	11	1996	2007	2	Seq # 4-11 n/a prior to 1/1/1998; Sensors 9 and 11 dead 2001.
37	1	11	1996	2007	0	Seq # 4-11 n/a prior to 1/1/1998.
38	1	11	1996	2007	0	Seq # 6-11 n/a prior to 1/1/1998.
39	4	42	1996	2007	9	Seq # 5-11, 15-22, 27-33, and 36-42 n/a prior to 1/1/1998; Seq # 36-42 removed 5/31/2005.
40	1	11	1996	2007	0	Seq # 5-11 n/a prior to 1/1/1998.
52	2	24	2000	2007	0	Seq # 1-16 came online 6/30/2000.
53	2	16	2000	2007	0	Seq # 17-32 came online 6/30/2000.
54	4	16	2000	2007	0	Seq # 1-16 came online 12/26/2004.
60	2	14	2004	2007	0	Seq # 1-14 came online 10/25/2004.
62	2	14	2005	2007	0	Seq # 1-14 came online 1/8/2005.
93	2	14	1998	2004	14	Seq # 1 removed 1/1/1999; Sensors 7 and 8 never came online; Seq # 2-6 and 9-14 removed 9/9/2004.
95	2	12	1998	2004	12	removed 9/9/2004.
97	2	16	1998	2007	7	Large gaps in some sensors.

[a] As of 2007

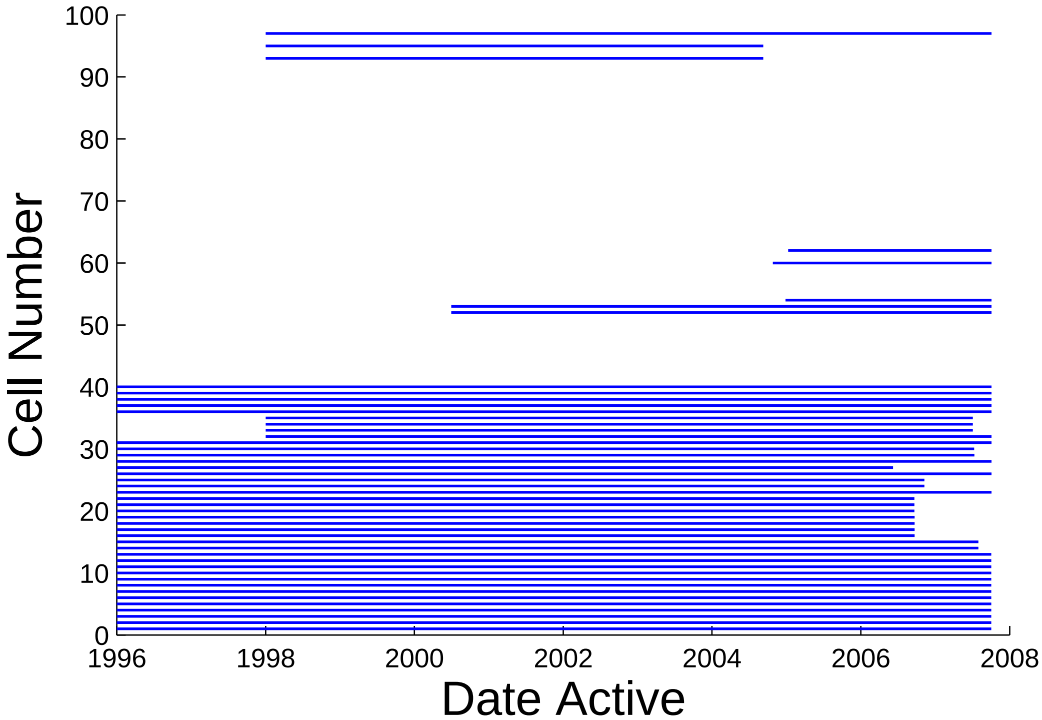

Figure 2.1: The active dates for the 48 MnROAD cells considered in this project are represented by the blue bars.

2.3 An Example: Cell #10, Sensor 3

Consider, for example, Cell #10. The engineered section for Cell #10 is shown in Figure 2.2.

Cell #10 has 4 sensor groups, comprising 42 individual sensors, with considered data spanning from 1996 through 2007. Sensors 6-11, 17-22, 28-33, and 36-42 were not available prior to 1 January 1998; however, all 42 sensors were still operational as of Fall 2007.

Continuing with this example, we focus on Cell #10, Sensor 3, Year 2001 (see Table 2.4).

There are $397,608$ considered temperature records for Cell #10, Sensor 3; see Figure 2.3. The annual (seasonal) variation in temperatures dominates this plot. The annual range in temperatures is from approximately (°C-20) to (°C40). The diurnal variations in temperature is hidden in this plot by the vast number of measurements and the size of the plotting symbol on the page. Note the visible gap in the data record[1] that occurs in Spring 2002.

There are $34,113$ temperature records for Cell #10, Sensor 3 during 2001. With a 15 minute data collection interval one would anticipate: 4 measurements per hour $\times$ 24 hours per day $\times$ 365 days $=$ 35,040 observations. A plot of the measured temperature data for this sensor, see Figure 2.4, shows the anticipated diurnal variation, which appears as jagged squiggles (vertical lines) in the figure, and the recognizable annual/seasonal variation, which causes the dominant sinusoidal shape.

Zooming in on a few days in the summer of 2001, see Figure 2.5, emphasizes the diurnal

[1]This gap is the result of using 0 as the designator for missing data. This issue is discussed at length in Section 6.4.

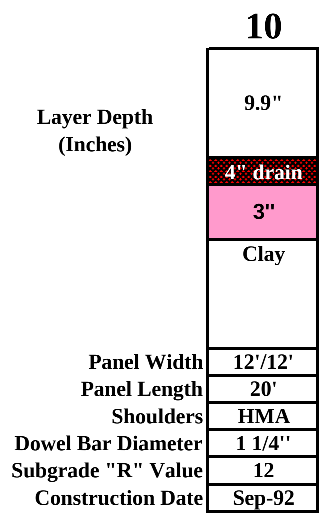

Figure 2.2: The vertical section, and associated engineering parameters, for Cell #10 is shown diagrammatically.

Table 2.4: Cell # 10, Sensor 3 is an example of a typical, well-behaved, temperature record. This table summarizes the associated data for 1 January 2001 through 31 December 2001.

Field	Value	Notes
Year	2001	
Count	34,113	Number of temperature measurements recorded.
Station	117046.9	
Offset	0.0030 m (0.01 ft)	Measured from the centerline of roadway.
Depth	0.1524 m (0.50 ft)	Measured as the distance below top of pavement.
Minimum	−15.9°C	Minimum temperatures from the data subset.
Maximum	40.9°C	Maximum temperatures from the data subset.
Median	11.4°C	Median temperatures from the data subset.

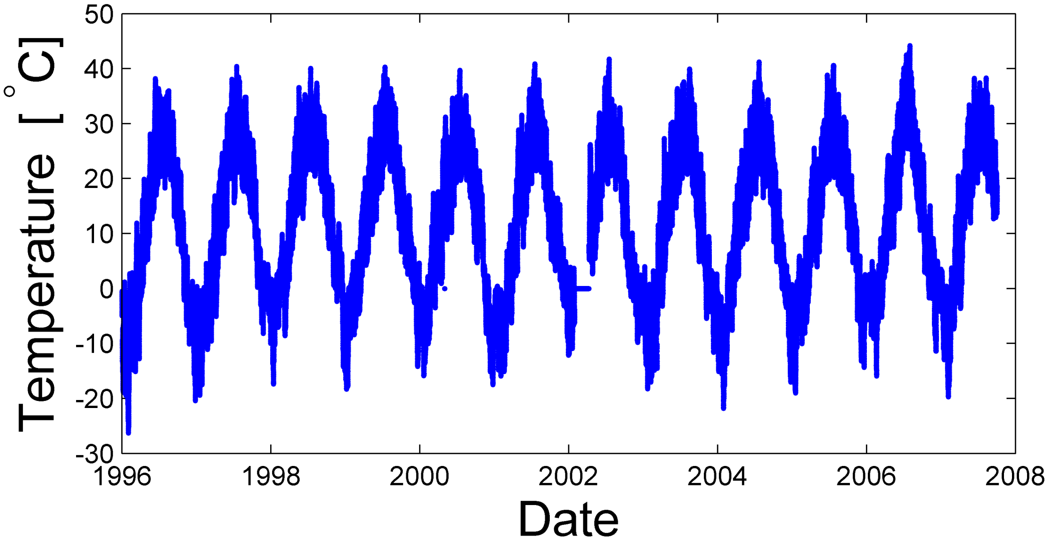

Figure 2.3: This is an example of a typical, well-behaved, temperature record. Shown are 397, 608 raw temperature measurements from Cell #10, Sensor 3. The anticipated annual variation of temperatures is clearly represented by the sinusoidal behavior of the plot.

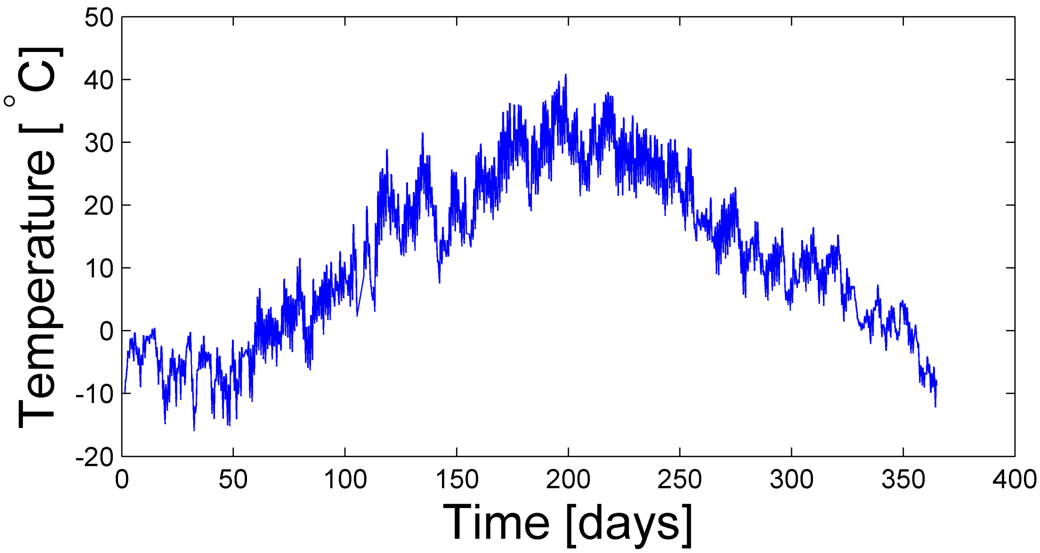

Figure 2.4: This is a typical example subset of the considered temperature data. Shown are 34, 133 raw measurements for Cell #10, Sensor 3. These data were collected from 1 January 2001 through 31 December 2001, with a 15 minute collection interval.

variation in the data, but the 285 measurements shown do not depict the annual/seasonal variation that dominates Figure 2.4.

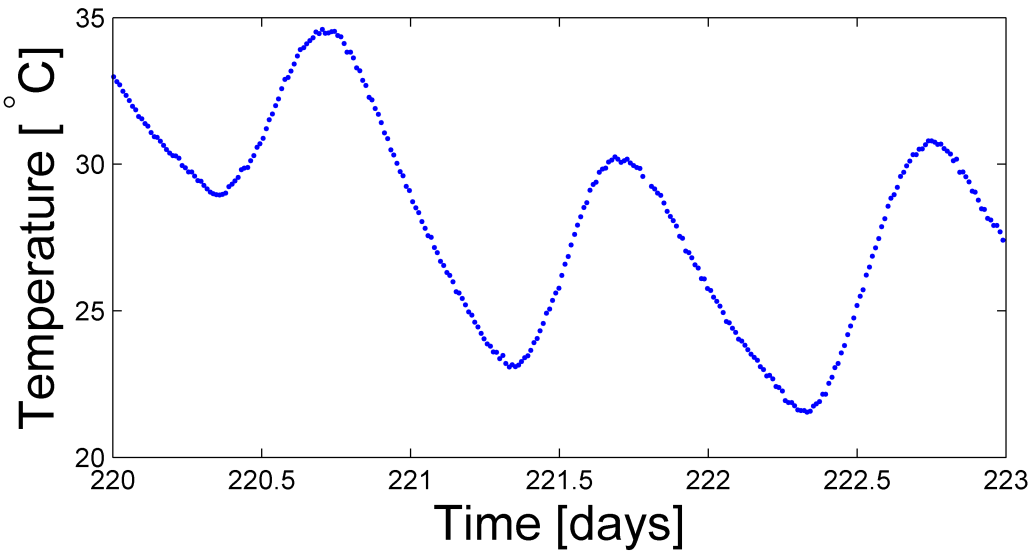

Figure 2.5: The anticipated diurnal variation is emphasized in this temperature data plot for a few days from Cell #10, Sensor 3. Shown are 285 raw measurements taken in the summer of 2001, with a 15 minute collection interval.

The $397,608$ considered temperature records from Cell #10, Sensor 3 span 592 weeks, from the Winter of 1996 through the Fall of 2007. Figures 2.6 and 2.7 show the weekly average and weekly standard deviation of the measured temperatures. As expected, the weekly average temperatures mirror the annual (seasonal) sinusoidal variation shown in 2.3. Interestingly, the weekly standard deviations show an annual sinusoidal variation as well: winters are much less variable than summers. Note that the gap in Spring 2002, which was identified in Figure 2.3, causes a visibly obvious perturbation in the weekly standard deviations.

2.4 Raw Data

The raw MnROAD data was supplied as a collection of 683 ASCII[2] text files using a *comma-separated values* format[3]. These raw data files were organized using a folder hierarchy. Every cell has its own folder. Every cell has a raw ASCII data file (.csv) for each year covered. In addition, every cell has a raw ASCII `SensorGroups` file.

Letting `RAW_ROOT` represent the root folder for the hierarchy, the raw data structure is organized as shown below.

```
RAW_ROOT
    Cell1
```

[2]ASCII is an acronym for *American Standard Code for Information Interchange*. See, for example, [43].

[3]The *comma-separate values* format, which is designated with a `.csv` filename extension, is a common format for tabular data (see [44]). This format is regularly used to export data from a Microsoft Excel spreadsheet.

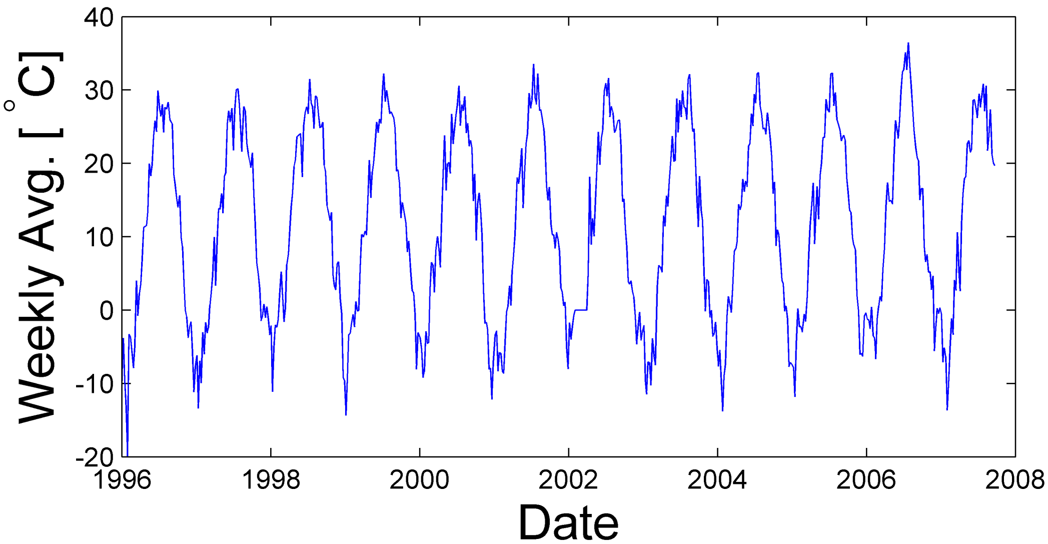

Figure 2.6: The anticipated annual variation is emphasized in this temperature data plot for 592 weekly averages of raw measured temperatures from Cell #10, Sensor 3. The anticipated annual (seasonal) variation dominates the underlying trend, although weather variations can be seen.

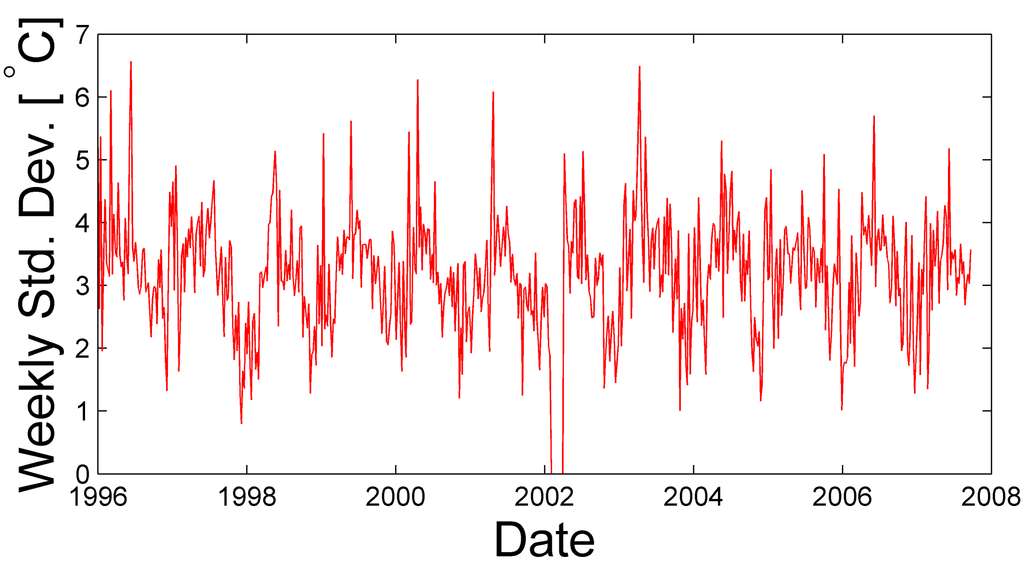

Figure 2.7: Systematic annual patterns in the temperature variability are apparent in the 592 weekly standard deviations of raw measured temperatures from Cell #10, Sensor 3. In general, winters are much less variable than summers.

```
        1996Cell1.csv
        1997Cell1.csv
        ...
        2007Cell1.csv
        Cell1SensorsGroups.csv
    Cell2
        1996Cell2.csv
        1997Cell2.csv
        ...
        2007Cell2.csv
        Cell2SensorsGroups.csv
    ...
```

This organization is clear, unambiguous, and extensible. However, the organization is inconvenient and a source of significant computational inefficiencies. Opening, reading, and closing hundreds of files, and processing 470 million temperature records stored using an ASCII representation is surprisingly slow.

2.5 Data Reorganization

The original raw text files are converted into a binary format. This conversion was necessary for computational speed. Converting from ASCII text to the MATLAB compressed binary format[4] increased the processing speed by two orders of magnitude in some cases. An additional benefit of this ASCII to binary conversion is the significant compression of the data files. All of the ASCII files combined require more than 4.45 GBytes, while the equivalent compressed binary file requires only 1.38 GBytes.

Unlike the original raw data, the converted data was organized by cell only. Specifically, all years are combined to make a single data file for each cell. The conversion function, named `CreateMatFiles.mat`, is a stand-alone function to convert the provided ASCII .csv files into MATLAB-specific, binary, .mat files.

The conversion function also takes the "year/day/hour/minute" representation from the ASCII source file and converts the data into a MATLAB serial date number: i.e. MATLAB's `datenum` format[5]. This conversion produces one binary file containing three arrays for each source cell. The various years represented by the ASCII source files are combined into one file. The three arrays are:

- D: ($N \times 1$) array of dates using MATLAB's `datenum` format,
- T: ($N \times S$) array of temperature data,

[4]Although the MATLAB platform offers many different file formats [14], the default `save` and `load` commands in MATLAB use compressed binary format [15]. These compressed binary files are designated with a `.mat` filename extension.

[5]Time is stored in decimal days from a nominal 1 January 0000. In MATLAB this is called the '`datenum`' format. This format can easily be converted into a standard calendar output using built in commands like `datestr`. See [13] for details. Note, the serial date number format for MATLAB differs from the serial date number format used by Microsoft Excel. Microsoft Excel uses 1 January 1900 as the starting date.

- F: (N × S) array of 16-bit unsigned integer flags[6].

where N is the total number of observations for the cell (the sum over all of the years) and S is the number of sensors in the cell. Note: there is one flag for each temperature record. Table 2.5 shows the conceptual organization of the converted data files.

Table 2.5: The converted data files include three parallel matrices: D, T, and F. This table shows the conceptual organization of the converted data files.

Example Cell								
D	T				F			
datenum	**Sensor 1**	**Sensor 2**	**Sensor 3**	**Sensor 4**	**F(1)**	**F(2)**	**F(3)**	**F(4)**
729025.006	26.8245	28.1523	30.2823	31.1485	0	0	0	0
729025.014	25.6848	27.4684	39.1648	30.6815	0	0	16	0
729025.025	25.1681	27.6484	29.6463	10.4682	0	0	0	48
⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓

Note: the sensors are numbered by the column of the 'T' matrix in which they reside. This numbering scheme parallels the numbering in the original raw ASCII data files: the first column in the source data file is designated Sensor 1, the second column is Sensor 2, and so forth.

[6]See Section 6.2 for more details on the interpretation and use of binary flags.

Chapter 3

Bestiary of Problems

3.1 Motivation

The MnROAD database consider in this project contains more than 471 million temperature records generated by 1,313 sensors over the twelve year period: 1996-2007. 340,361,568 of these temperature records (72.2%) are associated with active sensors. 324,429,599 temperature records pass all tests and are identified as reliable data after the analysis is complete. Thus, 15,931,969 temperature records from the active sensors are labeled as incorrect or unreliable (4.7%), and 146,748,725 temperature records are labeled as inactive, incorrect, or unreliable (31.1% of the total).

There are 146 million possible explanations for the inactive, incorrect, or unreliable data, but a small set of exceptional behaviors emerged from the analysis. This section presents a "bestiary" of example erroneous data behavior as a reference tool in the following presentation. These examples are meant to be demonstrative and not exhaustive.

The figures in this section are formatted slightly differently from other figures in this report. The date axis (abscissa) is formatted as decimal days from the nominal day zero (1 January 1996), rather than as calendar dates. This format is selected to highlight the relative time scales for each of the represented "beasts" occupying the bestiary.

3.2 Not-Yet-Operational Sensor

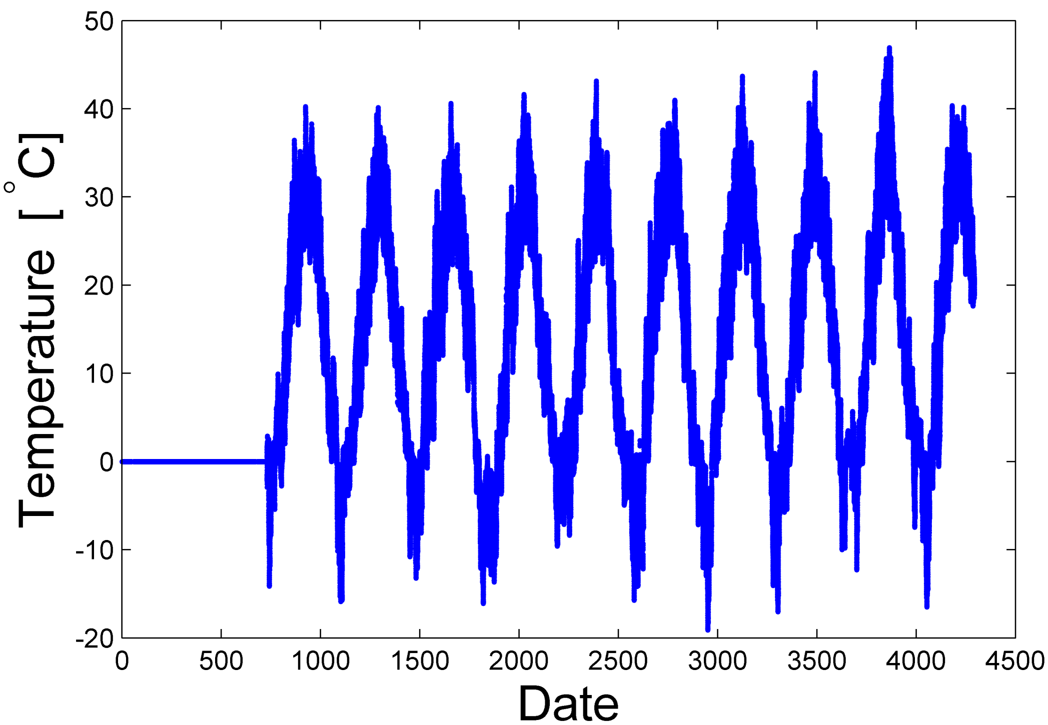

Figure 3.1: Cell #1, Sensor 4 is shown. Prior to day 700 (approximately), the sensor has not yet begun to record temperature data.

3.3 Deactivated Sensor

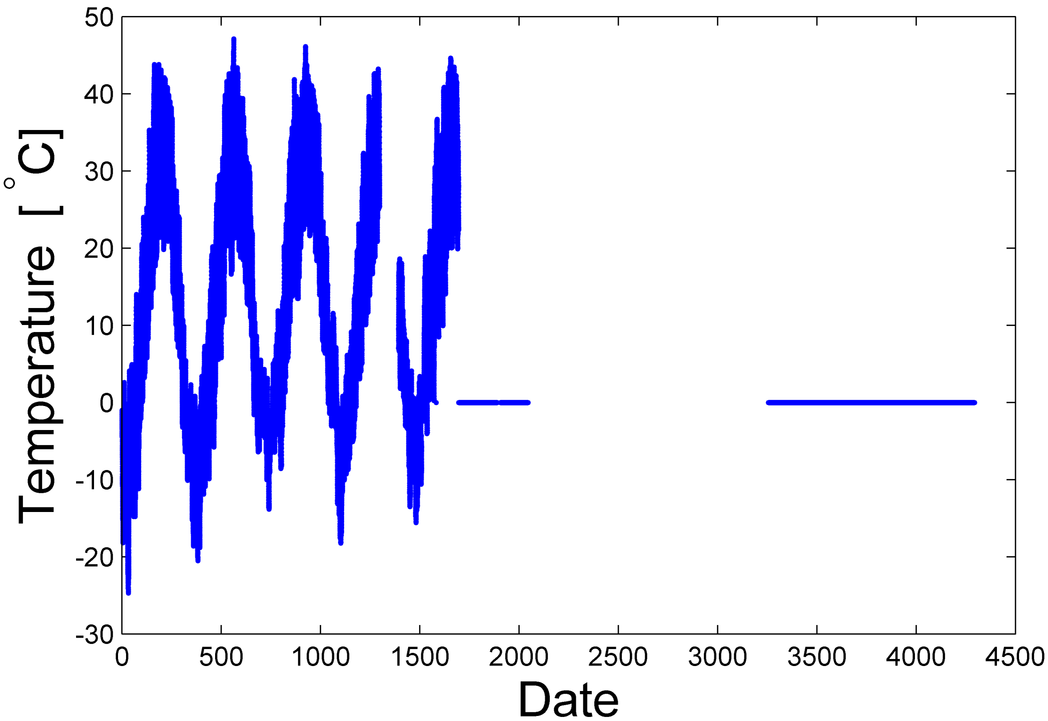

Figure 3.2: Cell #26, Sensor 2 is shown. After day 1, 700 (approximately), the sensor has stopped recording data indefinitely.

3.4 Data Gaps

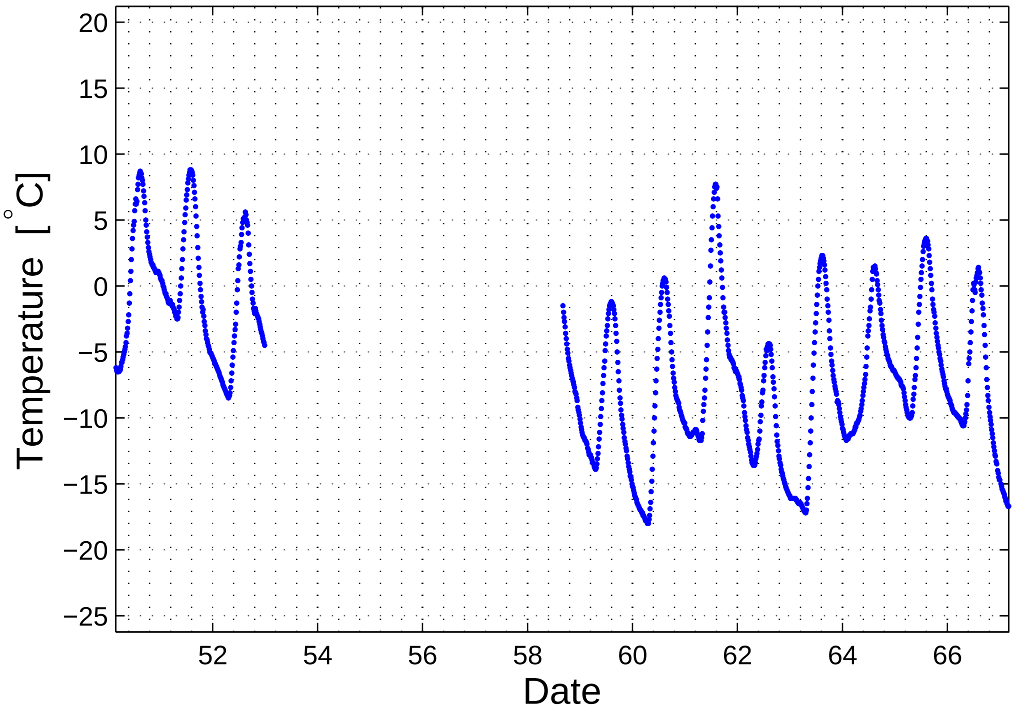

Figure 3.3: Cell #1, Sensor 1 is shown. There is a significant interval of time (about five days in this case) without any data recorded; e.g. the sensor or data collection device was turned off. While this is not inherently problematic, too many of such intervals indicate a faulty sensor or data collection device.

3.5 Data Skips

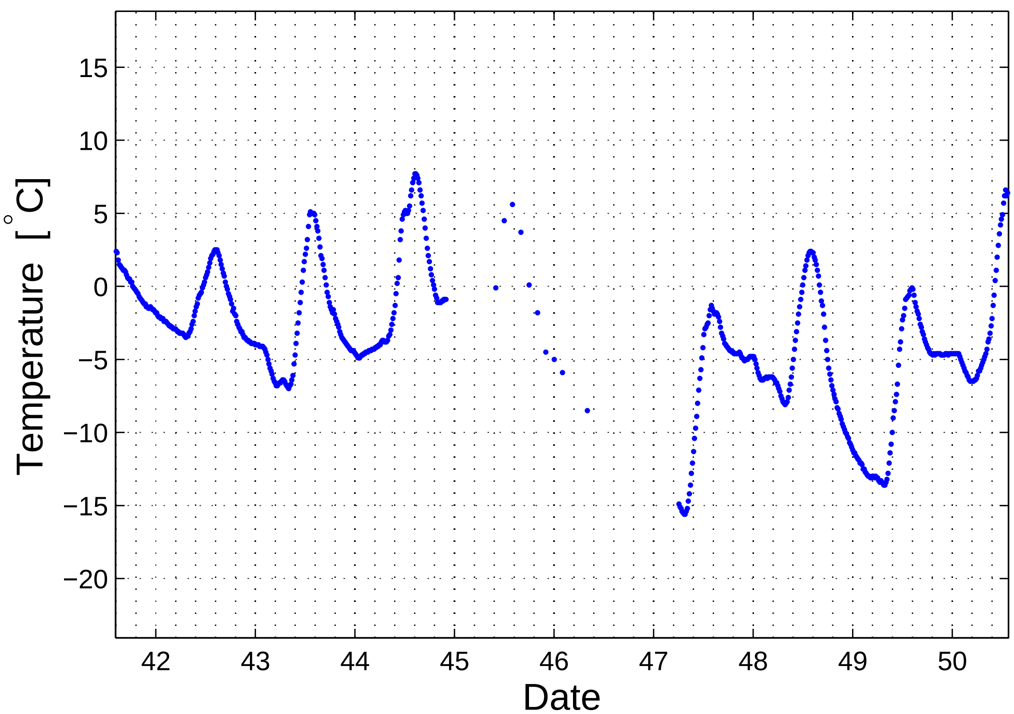

Figure 3.4: Cell #1, Sensor 1 is shown. There is a stretch of data (day 45 through the beginning of day 47) that follows the daily pattern, but at a much lower frequency than is represented by the rest of the time series. The data through this stretch are not exceptionally high or low, but they are unreliable.

3.6 Dead Data

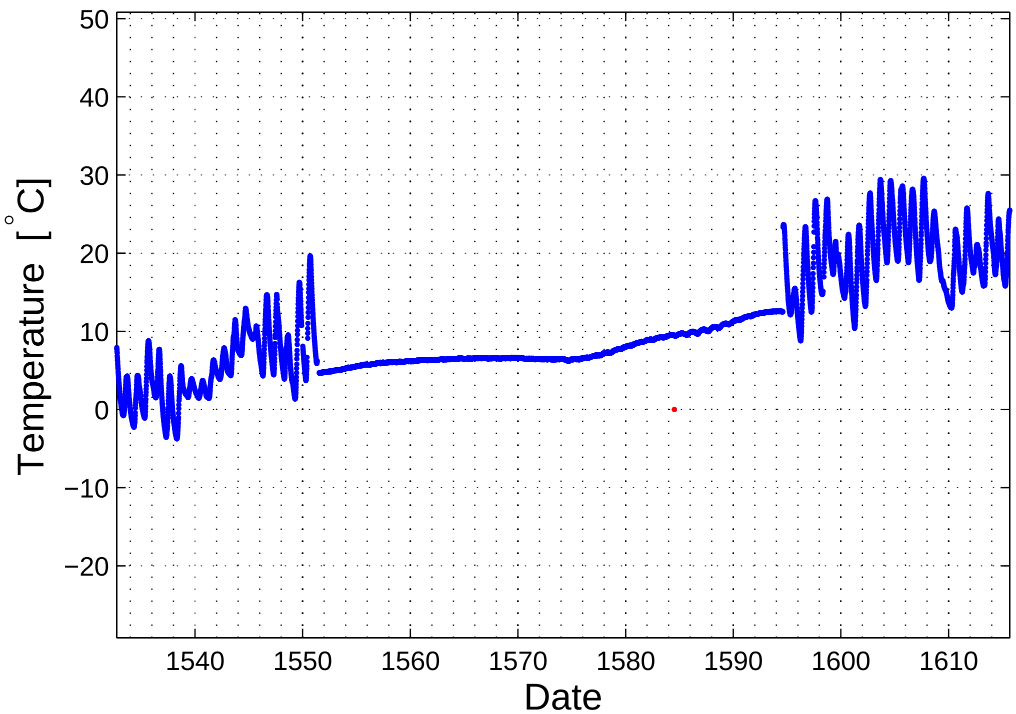

Figure 3.5: Cell #15, Sensor 3 is shown. The data between days 1,551 and 1,593 appears unnaturally flat, when compared to the surrounding normal diurnal patterns. This under-variation is visually obvious, but very difficult to catch as the individual values themselves are not extreme – not highly deviant from the surrounding data points. The key point of this problem is the lack of daily variation.

3.7 Data Hops

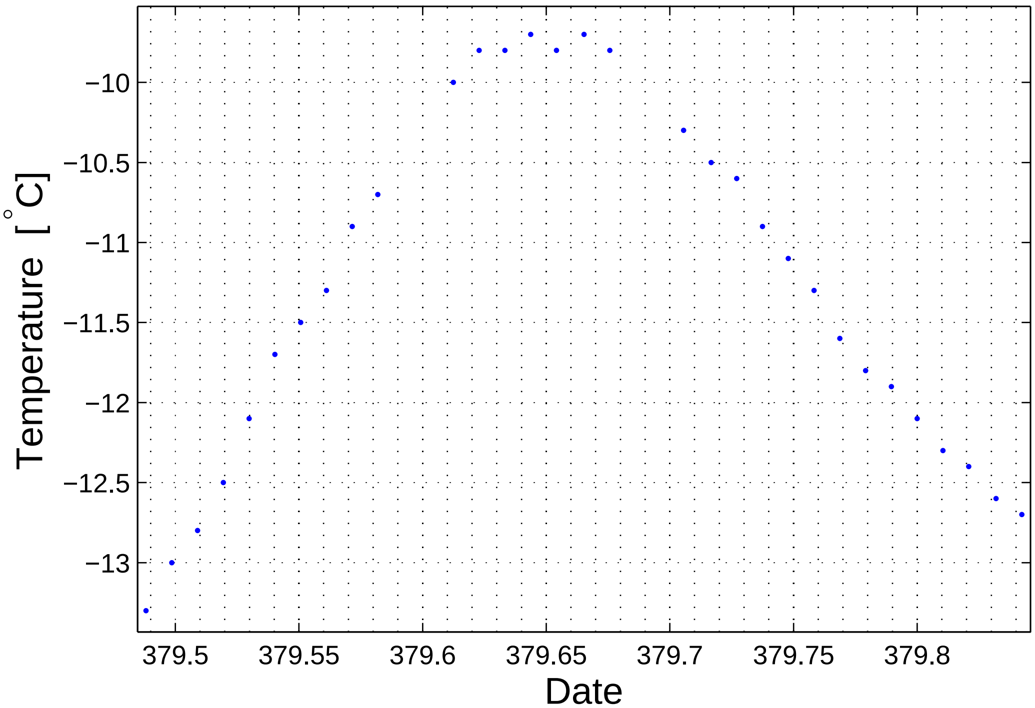

Figure 3.6: Cell #15, Sensor 3 is shown. This error is probably the simplest and least worrying of the erroneous data types. It occurs when a small amount of data is missing (e.g. at day 379.6, but the surrounding readings follow a sensible pattern. It is good data with a few points missing.

3.8 Outlying Data

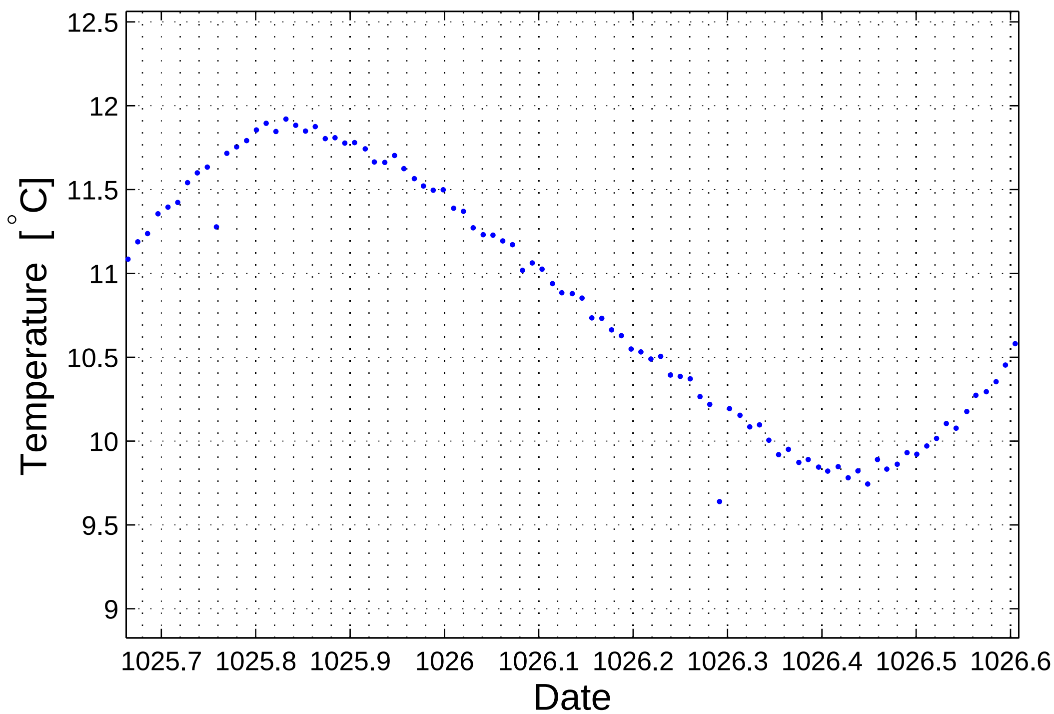

Figure 3.7: Cell #32, Sensor 1 is shown. This problem consists of data that fall within a realistic temperature change, but at an outlying position from the surrounding points.

3.9 Defective Sensor

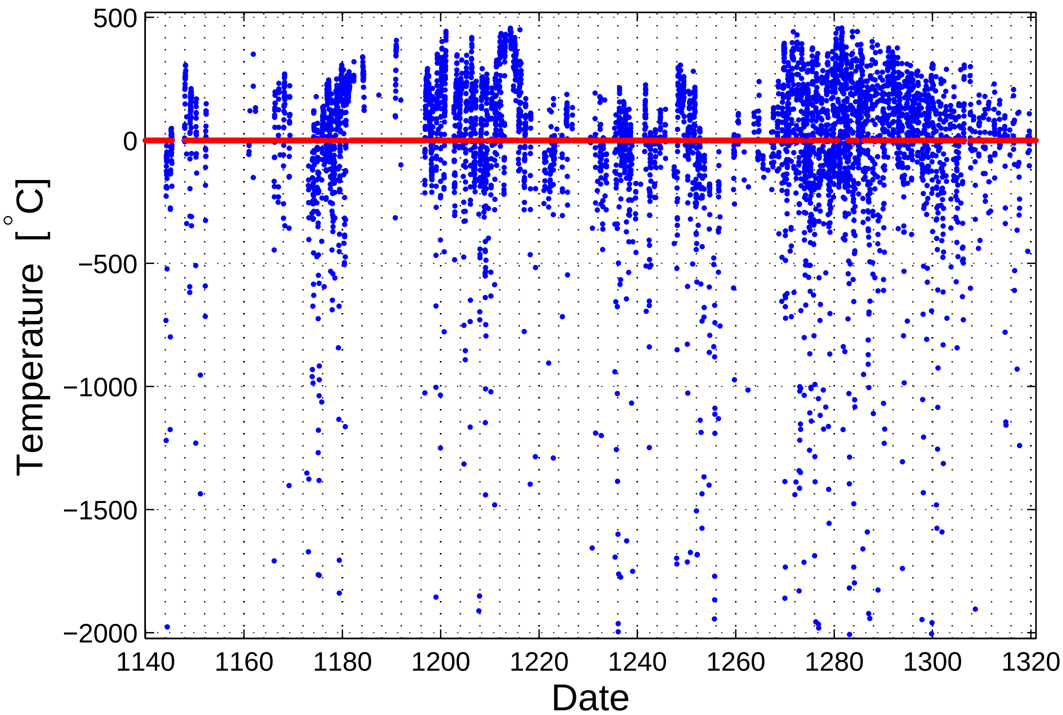

Figure 3.8: Cell #93, Sensor 1 is shown. The sensor readings follow no pattern and are clearly nonsensical. The only course of action is to disregard the data and retire the sensor.

3.10 Data Shift

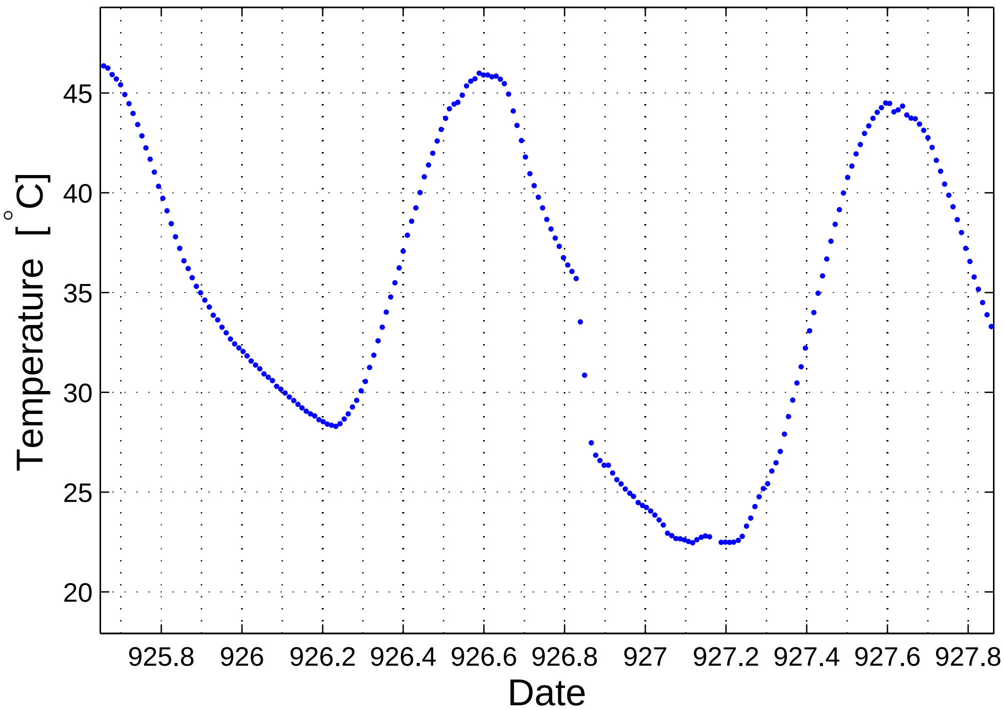

Figure 3.9: Cell #40, Sensor 1 is shown. The data shows an unrealistic, abrupt, change in temperature – dramatically deviating from the underlying sinusoidal trend. Often, this appears as though the sensor is suddenly calibrated differently. A possible explanation for such behavior, in shallow sensors during summer months, is a passing late-afternoon rainstorm.

3.11 Lost Data

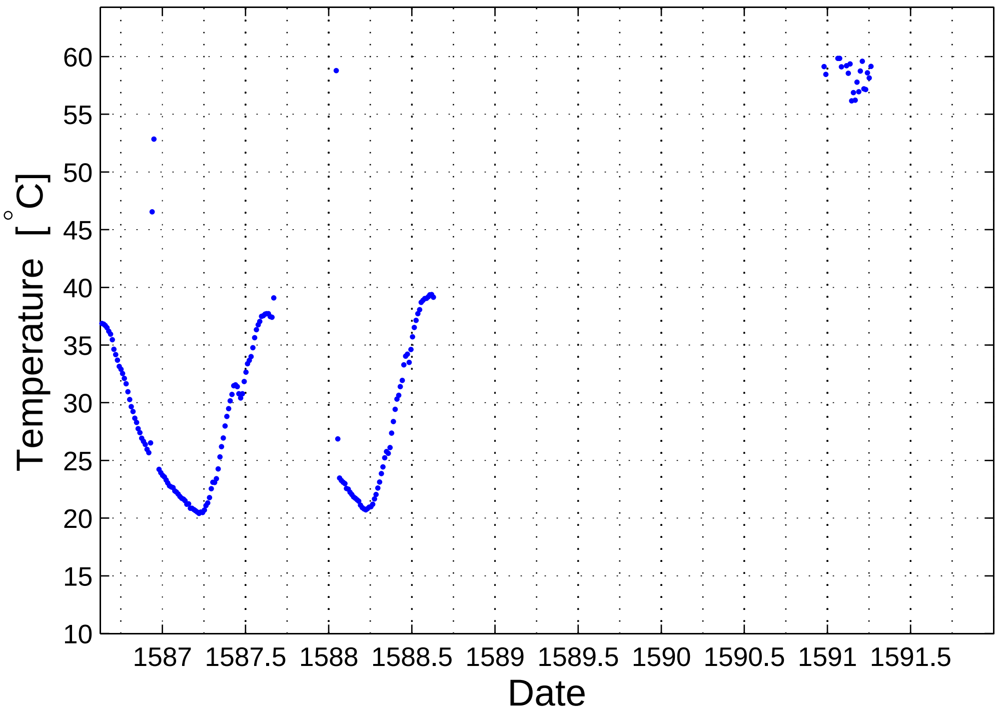

Figure 3.10: Cell #10, Sensor 1 is shown. The data shows an unrealistic change in temperature, as well as outlying from the surrounding trend.

Chapter 4

Data Analysis

4.1 Identification

The methods used to identify incorrect or unreliable data can be conveniently partitioned into four general categories.

1. **Sanity Check:** The sanity checks are not statistically based. Rather, the sanity checks are engendered by knowledge of and experience with the underlying processes. For example, a temperature measurement less than $-50°C$ or greater than $60°C$ is inconsistent with the thermal extremes in Minnesota – it is incorrect. Such is the case in Cell #93, Sensor 1; see Figure 3.8.

2. **Time Series Outliers:** The time-series-based checks are statistical in nature. These checks treat the stream of data generated by each sensor as an isolated univariate time series. The threshold values used to identify outliers are determined on a sensor-by-sensor basis using only data and statistics from the individual sensors.

3. **Group Outliers:** The group-based checks are statistical in nature. These checks compare specific computed values from individual sensors (like daily minimum, daily maximum, and daily range) to threshold values computed from statistical analysis of the data from all similar sensors[1].

4. **Intermittency:** The intermittency checks are not statistically based. These checks are applied after all of the the other checks have been completed. These checks identify unreliable data – even though they have not been flagged by one of the previous checks. For example, data from a sensor that fails one or more of the previous checks throughout the morning should be questioned in the afternoon. Another example would be a sensor that produces and hour of "bad data", then an hour of "good data", then an hour of "bad data", etc. Even though the "good data" is not flagged by one of the previous checks, it must be questioned since the sensor is unreliable.

[1] Similar sensors have the same pavement type, the same location (Mainline or Low Volume Road), and the same approximate depth. The determination and grouping of "similar sensors" is discussed in Section 4.2.

4.2 Sensor Subsets and Grouping

In order to analyze sensor data adequately, it is important to compare it to the data from other similar sensors. The question then becomes, which sensors are comparable? Assuming that the sensor temperature is a function of the surface temperature, the pavement type, the sensor's location (Mainline or Low Volume Road) and the sensor's depth, sensors can be compared to others with similar depths, locations and pavement types.

1. The first step in organizing sensors was to split the sensors into two subsets by pavement type - asphalt in one, concrete in the other. This is an extremely uncomplicated classification. The type criterion is denoted by either a 1 (asphalt) , or a 2 (concrete).

2. The majority of the sensors were already put into one of thirteen groups depending on their depth. The median depth of each depth group was calculated and used to assign groups to the unaffiliated sensors. Every unaffiliated sensor was assigned to the depth group with the median depth closest to the sensor's depth. This was performed for both of the pavement types. Although the preexisting groupings have some overlap in heights, the usefulness of this subset remains. Figures 4.1 and 4.2 show the final groups. The location criterion is denoted by the numbers 1-13 (13 being the deepest group).

3. Lastly, each pavement type subset has some sensors on the Mainline and some on the Low Volume Road. The sensors are then classified by their location type. The location criterion is denoted by either a 1 (Mainline) , or a 2 (Low Volume Road).

Once every sensor has three criteria, error recognition can be greatly increased. When comparing two sensors, they must share all three groups. For example, a sensor that is in depth group 10, has asphalt pavement type (denoted as `pavetype`=1), and on the mainline (also denoted as 1), can only be compared with another sensor that has a grouping of [10, 1, 1].

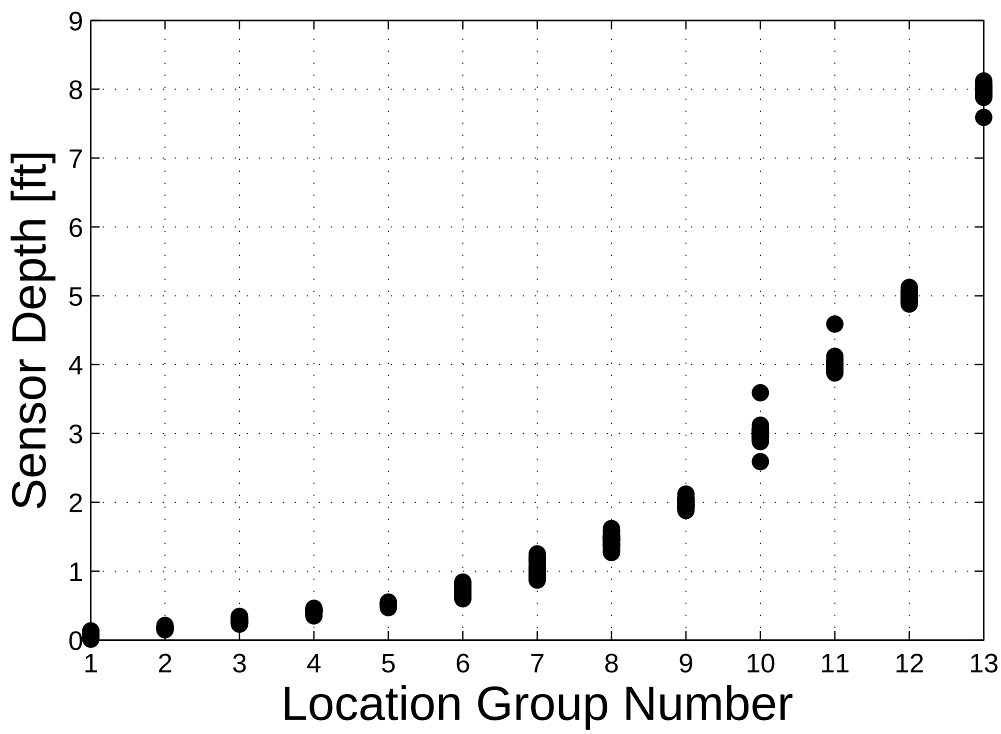

Figure 4.1: A black disk is plotted for each sensor in the HMA cells. This shows the range of sensor depths for each of the 13 Mn/DOT -defined Location Groups. There is some overlap between the groups.

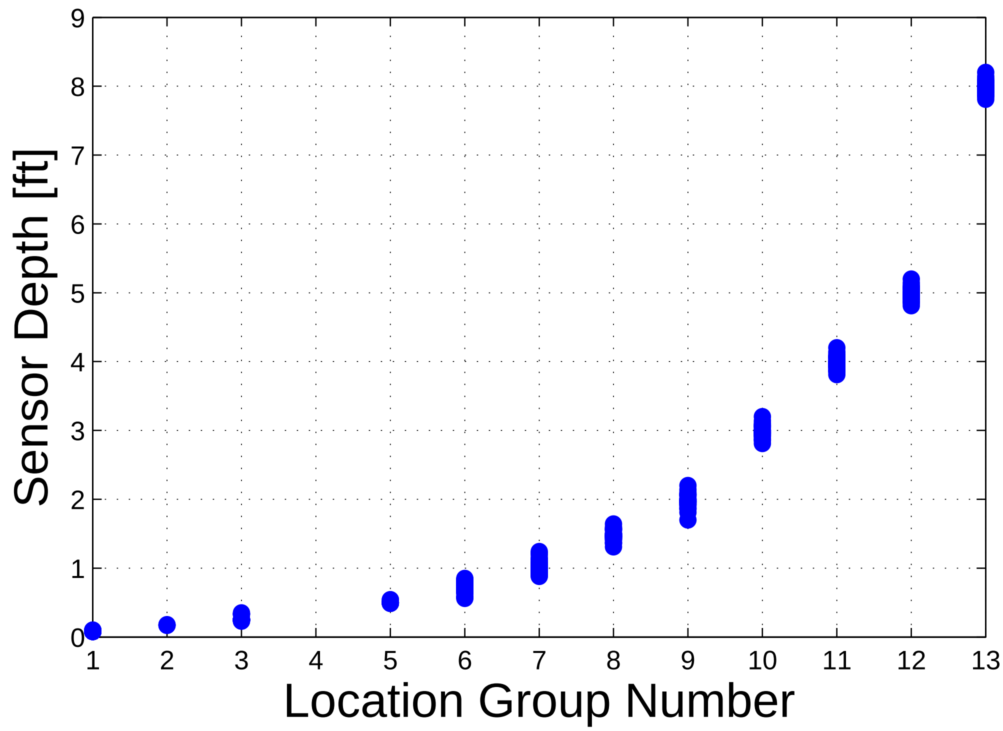

Figure 4.2: A blue disk is plotted for each sensor in the PCC cells. This shows the range of sensor depths for each of the 13 Mn/DOT -defined Location Groups. There is some overlap between the groups.

Chapter 5

Statistical Background

5.1 Moment-based Statistics

5.1.1 Mean

The mean, or expected value, of a random variable is commonly denoted μ. When the density function, $f(x)$, exists the mean is given by [2, 7, 20, 27, 35, 29, 28]

$$\mu = \int_{-\infty}^{+\infty} x f(x) dx \tag{5.1}$$

The mean is a measure of location for a distribution: a representative middle value. As shown by 5.1, the mean is the centroid of the density function.

The *arithmetic average* (or more simply, the average), is the most common estimator for the mean from a finite collection of observed values. Let $\{x_1, x_2, \ldots, x_n\}$ represent a collect of n observations, then the arithmetic average, denoted $\bar{x}$, is given by

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{5.2}$$

5.1.2 Variance

The variance of a random variable is commonly denoted σ^2. When the density function, $f(x)$, exists the variance is given by [2, 20, 7, 27, 42]

$$\sigma^2 = \int_{-\infty}^{+\infty} (x - \mu_X)^2 f(x) dx \tag{5.3}$$

The variance is a measure of spread for a distribution. As shown by 5.3, the variance is the rotational moment of inertia of the density function.

The *sample variance* is the most common estimator for the variance from a finite collection of observed values. Let $\{x_1, x_2, \ldots, x_n\}$ represent a collection of n observations, then the sample variance, denoted s^2, is given by

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \tag{5.4}$$

5.1.3 Standard Deviation

The standard deviation of a random variable is commonly denoted σ. When the density function, $f(x)$, exists the standard deviation is given by [2, 20, 7, 27, 41]

$$\sigma = \sqrt{\int_{-\infty}^{+\infty} (x - \mu)^2 f(x) dx} \tag{5.5}$$

The standard deviation is simply the square root of the variance. The standard deviation is a measure of spread for a distribution: a representative scale.

The *sample standard deviation* is most common estimator for the standard deviation from a finite collection of observed values. Let $\{x_1, x_2, \ldots, x_n\}$ represent a collect of n observations, then the sample standard deviation, denoted s_X, is given by

$$s_X = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{5.6}$$

5.1.4 Skewness

The skewness of a random variable is commonly denoted γ_1. When the density function, $f(x)$, exists the skewness is given by [2, 40]

$$\gamma_1 = \frac{\int_{-\infty}^{+\infty} (x - \mu)^3 f(x) dx}{\sigma^3} \tag{5.7}$$

The skewness is a measure of shape for a distribution. More specifically, the skewness measures asymmetry of the density function.

The *sample skewness* is most common estimator for the skewness from a finite collection of observed values. Let $\{x_1, x_2, \ldots, x_n\}$ represent a collect of n observations, then the sample skewness, denoted g_1, is given by

$$g_1 = \frac{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^3}{s^3} \tag{5.8}$$

5.1.5 Kurtosis

The kurtosis of a random variable is commonly denoted γ_2. When the density function, $f(x)$, exists the kurtosis is given by [2, 32]

$$\gamma_2 = \frac{\int_{-\infty}^{+\infty} (x - \mu)^4 f(x) dx}{\sigma^4} - 3 \tag{5.9}$$

This is also known as the *excess kurtosis*. The kurtosis is a measure of shape for a distribution. More specifically, the kurtosis measures peakedness of the density function.

The *sample kurtosis* is most common estimator for the kurtosis from a finite collection of observed values. Let $\{x_1, x_2, \ldots, x_n\}$ represent a collect of n observations, then the sample kurtosis, denoted g_2, is given by

$$g_2 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{s^4} - 3 \tag{5.10}$$

5.2 Order-based Statistics

5.2.1 Median

The median of a random variable is the 50th percentile. When the density function, $f(x)$, exists the median is defined to be the value m such that [2, 20, 7, 27, 5, 36]

$$\int_{-\infty}^{m} f(x)dx = \int_{m}^{+\infty} f(x)dx = 0.5 \tag{5.11}$$

The median is a measure of location for a distribution: a representative middle value. The median and the mean are equal only when the distribution is symmetric.

The *sample median* is the most common estimator for the median from a finite collection of observed values. Let $\{x_1, x_2, \ldots, x_n\}$ represent a collection of n observations, then the sample median is computed by first sorting the data into ascending order:

$$\{x_{[1]}, x_{[2]}, \ldots, x_{[n]}\} \tag{5.12}$$

where the subscript in brackets, e.g. [1], denotes the sort order. The sample median, denoted $\tilde{x}$, is then given by[1]

$$\tilde{x} = x_{[\lceil n/2 \rceil]} \tag{5.13}$$

where $\lceil n/2 \rceil$ denotes the smallest integer not less than $n/2$.

5.2.2 Lower Quartile

The lower quartile of a random variable is the 25th percentile. When the density function, $f(x)$, exists the lower quartile is defined to be the value lq such that [21, 26, 39]

$$\int_{-\infty}^{lq} f(x)dx = 1 - \int_{lq}^{+\infty} f(x)dx = 0.25 \tag{5.14}$$

[1]The median is a simple concept: the 50th percentile. Nonetheless, the computation of an estimate of a specific percentile from a finite collection of observations has no generally agreed upon standard formula [38]. For example, if the desired percentile does not align exactly with an observation (as is the case when computing the 50th percentile of an even number of observations), should an interpolated value be used? Such an interpolation is recommended by Prins et al. [24]. In this work, the simpler "rounding up" definition (see 5.13, 5.16, 5.19) is consistently applied. For large data set, the difference is insignificant.

The lower quartile is also known as the lower hinge.

The *sample lower quartile* is the most common estimator for the lower quartile from a finite collection of observed values. Let $\{x_1, x_2, \ldots, x_n\}$ represent a collect of n observations, then the sample lower quartile is computed by first sorting the data into ascending order:

$$\{x_{[1]}, x_{[2]}, \ldots, x_{[n]}\} \tag{5.15}$$

where the subscript in brackets, e.g. [1], denotes the sort order. The sample lower quartile, denoted x_{lq}, is then given by

$$x_{lq} = x_{[\lceil 0.25n \rceil]} \tag{5.16}$$

where $\lceil 0.25n \rceil$ denotes the smallest integer not less than $0.25n$.

5.2.3 Upper Quartile

The upper quartile of a random variable is the 75th percentile. When the density function, $f(x)$, exists the upper quartile is defined to be the value uq such that [21, 26, 39]

$$\int_{-\infty}^{uq} f(x)dx = 1 - \int_{uq}^{+\infty} f(x)dx = 0.75 \tag{5.17}$$

The upper quartile is also known as the upper hinge [21, 26].

The *sample upper quartile* is the most common estimator for the upper quartile from a finite collection of observed values. Let $\{x_1, x_2, \ldots, x_n\}$ represent a collection of n observations, then the sample upper quartile is computed by first sorting the data into ascending order:

$$\{x_{[1]}, x_{[2]}, \ldots, x_{[n]}\} \tag{5.18}$$

where the subscript in brackets, e.g. [1], denotes the sort order. The sample upper quartile, denoted x_{uq}, is then given by

$$x_{uq} = x_{[\lceil 0.75n \rceil]} \tag{5.19}$$

where $\lceil 0.75n \rceil$ denotes the smallest integer not less than $0.75n$.

5.2.4 Inter-quartile Range

The inter-quartile range of a random variable is the distance from the 25th percentile to the 75th percentile. When the density function, $f(x)$, exists the inter-quartile range is defined as [21, 26, 31]

$$\text{inter-quartile range} = \text{upper quartile} - \text{lower quartile} \tag{5.20}$$

The inter-quartile range is a measure of spread: an internal scale. The inter-quartile range is also known as the hinge spread [21, 26]. The inter-quartile range captures the middle fifty percent of the population.

The *sample inter-quartile range* is the most common estimator for the inter-quartile range from a finite collection of observed values. Let $\{x_1, x_2, \ldots, x_n\}$ represent a collection of n observations, then the sample inter-quartile range is computed by first sorting the data into ascending order:

$$\{x_{[1]}, x_{[2]}, \ldots, x_{[n]}\} \tag{5.21}$$

where the subscript in brackets, e.g. [1], denotes the sort order. The sample inter-quartile range, denoted $\mathtt{iqr}$,is then given by

$$\mathtt{iqr} = x_{uq} - x_{lq} \tag{5.22}$$

The sample inter-quartile range captures the middle fifty percent of the observed values. The inter-quartile range is also known as the *hinge spread* and denoted H.

5.2.5 Lower-quartile Spread

The lower-quartile spread of a random variable is the distance from the 25th percentile to the median. When the density function, $f(x)$, exists the lower-quartile range is defined as

$$\text{lower-quartile spread} = \text{median} - \text{lower quartile} \tag{5.23}$$

The lower-quartile spread is a measure of spread that allows for asymmetry.

The *sample lower-quartile spread* is the most common estimator for the lower-quartile spread from a finite collection of observed values. Let $\{x_1, x_2, \ldots, x_n\}$ represent a collection of n observations, then the sample lower-quartile spread is computed by first sorting the data into ascending order:

$$\{x_{[1]}, x_{[2]}, \ldots, x_{[n]}\} \tag{5.24}$$

where the subscript in brackets, e.g. [1], denotes the sort order. The sample lower-quartile spread, denoted $\mathtt{lqs}$, is then given by

$$\mathtt{lqs} = \tilde{x} - x_{lq} \tag{5.25}$$

The sample lower-quartile spread captures the second quarter of the data.

5.2.6 Upper-quartile Spread

The upper-quartile spread of a random variable is the distance from the median to the 75th percentile. When the density function, $f(x)$, exists the upper-quartile range is defined as

$$\text{upper-quartile spread} = \text{upper quartile} - \text{median} \tag{5.26}$$

The upper-quartile spread is a measure of spread that allows for asymmetry.

The *sample upper-quartile spread* is the most common estimator for the upper-quartile spread from a finite collection of observed values. Let $\{x_1, x_2, \ldots, x_n\}$ represent a collect of n observations, then the sample upper-quartile spread is computed by first sorting the data into ascending order:

$$\{x_{[1]}, x_{[2]}, \ldots, x_{[n]}\} \tag{5.27}$$

where the subscript in brackets, e.g. [1], denotes the sort order. The sample upper-quartile spread, denoted $\mathtt{uqs}$, is then given by

$$\mathtt{uqs} = x_{uq} - \tilde{x} \tag{5.28}$$

The sample upper-quartile spread captures the third quarter of the data.

5.3 Distributional Models

One of the more useful types of statistical analysis in civil engineering is the fitting and comparing of data to theoretical distributional models. When a data set is described as *approximately Normal* or *approximately Lognormal* it conjures a mental image and suggests sets of potentially useful heuristics. For example, with a Normal distribution:

- the population is characterized by two parameters: mean and standard deviation;
- the histogram is symmetric about the mean;
- the mean equals median equals the mode;
- the distance from the axis of symmetry (the mean) to the inflection point of the histogram is one standard deviation;
- approximately two-thirds of the population is captured between the mean, plus and minus one standard deviation;
- approximately ninety-five percent of the population is captured between the mean, plus and minus two standard deviations;
- the tails die out quickly, but extreme values are possible on both ends; and
- the mean can be pretty well estimated by the average of 30 or 40 independent observations.

The purpose of such distributional models is not to dictate or constrain the state-of-nature, for real data in civil engineering never fits the theoretical distributional models exactly. Like elasticity with Portland cement concrete, and visco-plasticity with bituminous asphalt, theoretical distributions are models that – within an appropriate domain of application – serve pragmatic purposes.

Over the course of the MnROAD analysis presented in this report, four different theoretical distributions were found to be particularly useful:

- Normal (Gaussian) distribution;
- Lognormal distribution;
- Laplace Distribution (a special case of the Generalized Normal); and
- Generalized Normal Distribution.

Many other theoretical distributional models were considered and discarded as ill-fitting or uninformative.

5.3.1 Normal (Gaussian) Distribution

Parameters

The most common representation for the Normal distribution has two parameters:

μ the mean, which is a measure of location; and

σ the standard deviation, which is a measure of spread.

All other parametric statistics of the Normal distribution can be given in terms of μ and σ. See Table 5.1.

Table 5.1: Normal distribution parameters and notation.

mean	μ
median	μ
mode	μ
standard deviation	σ
variance	σ^2
skewness	0
kurtosis	3
lower quartile	$\mu - 0.6745\sigma$
upper quartile	$\mu + 0.6745\sigma$
inter-quartile range	1.3490σ

Probability Density Function

The probability density function for the Normal distribution is given by [10, 37]

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left[\frac{x-\mu}{\sigma}\right]^2\right) \tag{5.29}$$

for $-\infty < x < +\infty$. Equation (5.29) is nothing more than the analytic form of the well-known "bell shaped curve"; see Figure 5.1. Equation (5.29) is symmetric about $x = \mu$, and has inflection points at $x = \mu \pm \sigma$.

In MATLAB, the Normal distribution probability density function is given by the command `normpdf`. The equivalent command in Microsoft EXCEL is `normdist` (with the last argument set to "false").

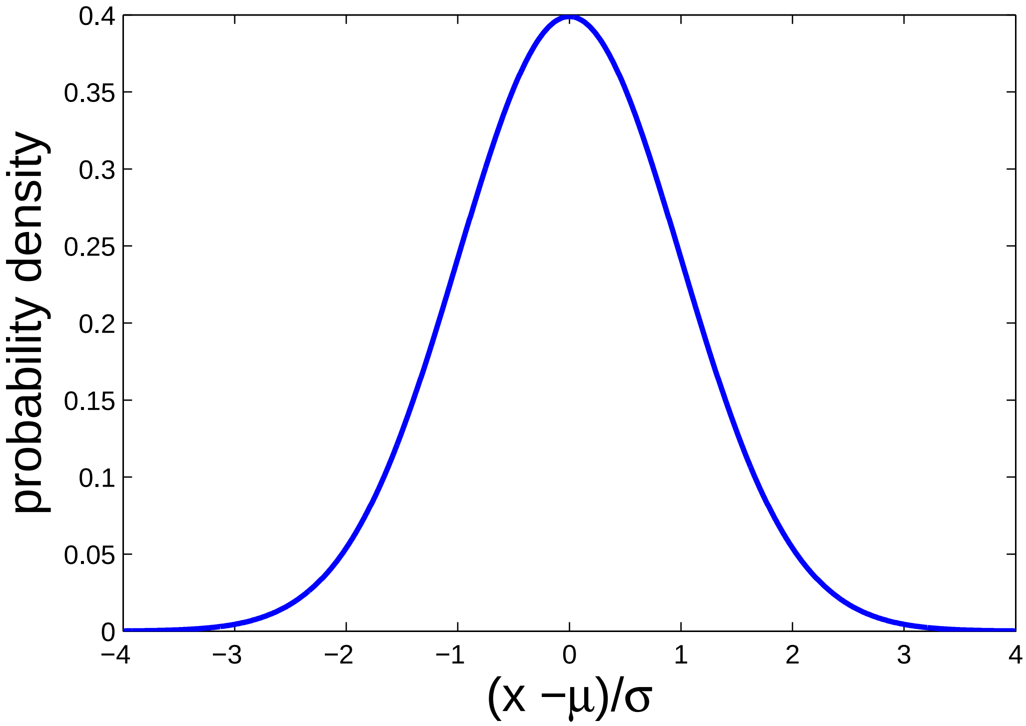

Figure 5.1: Normal distribution probability density function.

Cumulative Distribution Function

The cumulative distribution function for the Normal distribution cannot be expressed in terms of common elementary functions. Although we can formally write

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{1}{2}\left[\frac{\zeta-\mu}{\sigma}\right]^2\right) d\zeta \tag{5.30}$$

we can not actually do this integration. That is, we can not represent this integral as a finite combination of elementary functions. We can, however, represent the Normal distribution function as an infinite series; for $z \geqslant 0$ we have [46]:

$$F(x) = \frac{1}{2} + \frac{1}{\sqrt{2\pi}} \sum_{n=0}^{+\infty} \frac{(-1)^n z^{2n+1}}{n!\, 2^n\, (2n+1)} \tag{5.31}$$

where

$$z = \frac{x-\mu}{\sigma} \tag{5.32}$$

Equation (5.31) is mathematically correct, but computationally inefficient: the infinite series converges very, very slowly.

A useful rational approximation for the Gaussian distribution function is given by Abramowitz and Stegun [1, (26.2.18)]. For $z \geqslant 0$,

$$F(x) = 1 - \frac{1}{2}\left[\frac{1}{1 + c_1 z + c_2 z^2 + c_3 z^3 + c_4 z^4}\right]^4 + R(z) \tag{5.33}$$

where z is a normalized x, given by (5.32), and

$$c_1 = 0.196854 \qquad c_2 = 0.115194$$
$$c_3 = 0.000344 \qquad c_4 = 0.019527$$

The absolute value of the approximation error term, $R(z)$, is never greater than 0.00025.

Equation (5.33) is a relatively easy formula to compute and remarkably accurate. For quick-and-dirty computations with a hand-held calculator there is an even simpler approximation given by Abramowitz and Stegun [1, (26.2.24)]. For $z \geqslant 0$,

$$F(x) \approx \frac{1 + \sqrt{1 - \exp\left(-2z^2/\pi\right)}}{2} \tag{5.34}$$

where z is as given by (5.32). The magnitude of the error for (5.34) is never greater than 0.004, which is accurate enough for many engineering computations.

Notice that both approximations, (5.33) and (5.34), are defined for $z \geqslant 0$. To compute $F(x)$ for $z < 0$, you could take advantage of the symmetry of the Normal distribution and apply the identity

$$F(-x) = 1 - F(x) \tag{5.35}$$

In `MATLAB` the Normal cumulative distribution function is given by the command `normcdf`. The equivalent command in Microsoft `EXCEL` is `normdist` (with the last argument set to "true").

5.3.2 Lognormal Distribution

Parameters

Let X represent a random variable that follows a lognormal distribution, with mean (μ) and standard deviation (σ):

$$\mu = \boldsymbol{E}(X) \tag{5.36}$$

$$\sigma = \boldsymbol{Std}(X) \tag{5.37}$$

Let Y represent the random variable derived from X by taking the natural logarithm $Y = \ln X$. The mean and standard deviation of this log-transformed random variable, Y, will be denoted α and β:

$$\alpha = \boldsymbol{E}(\ln X) = \boldsymbol{E}(Y) \tag{5.38}$$

$$\beta = \boldsymbol{Std}(\ln X) = \boldsymbol{Std}(Y) \tag{5.39}$$

Reiterating: μ and σ are the mean and standard deviation of the lognormal random variable, while α and β are the mean and standard deviation of the log-transformed variable. The relationships between (μ, σ) and (α, β) are:

$$\mu = \exp\left(\alpha + \frac{\beta^2}{2}\right) \tag{5.40}$$

$$\sigma = \mu\sqrt{\exp(\beta^2) - 1} \tag{5.41}$$

Inverting (5.40) and (5.41) yields

$$\beta^2 = \ln\left(1 + \left[\frac{\sigma}{\mu}\right]^2\right) \tag{5.42}$$

$$\alpha = \ln(\mu) - \frac{\beta^2}{2} \tag{5.43}$$

Note that β is a function of the coefficient of variation of X, and not a function of the mean or standard deviation alone; thus, β is a dimensionless measure of variability. Further note that the expected value of the logarithm (α) does not equal the logarithm of the expected value (μ).

All other parametric statistics of the lognormal distribution can be given in terms of μ and σ, although it is often more convenient to use α and β. See Table 5.2.

Table 5.2: Lognormal distribution parameters and notation.

mean	μ
median	e^{α}
mode	$e^{\alpha-\beta^2}$
standard deviation	σ
variance	σ^2
skewness	$\left(e^{\beta^2}+2\right)\sqrt{e^{\beta^2}-1}$
kurtosis	$e^{4\beta^2}+2e^{3\beta^2}+3e^{2\beta^2}-3$
lower quartile	$e^{\alpha}/e^{0.6745\beta}$
upper quartile	$e^{\alpha}e^{0.6745\beta}$
inter-quartile range	$e^{\alpha}\left(e^{0.6745\beta}-1/e^{0.6745\beta}\right)$

Probability Density Function

Using the α and β notation, the two-parameter lognormal density function is given by [10, 34]

$$f(x) = \frac{1}{x\beta\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left[\frac{\ln x - \alpha}{\beta}\right]^2\right) \tag{5.44}$$

for $0 < x < +\infty$. See Figure 5.2.

In `MATLAB`, the lognormal distribution probability density function is given by the command `lognpdf`. There is no equivalent command in Microsoft `EXCEL`.

Cumulative Distribution Function

As with the normal distribution there is no closed form representation of the lognormal distribution function using elementary functions – we must rely on tables and approximations. The cumulative

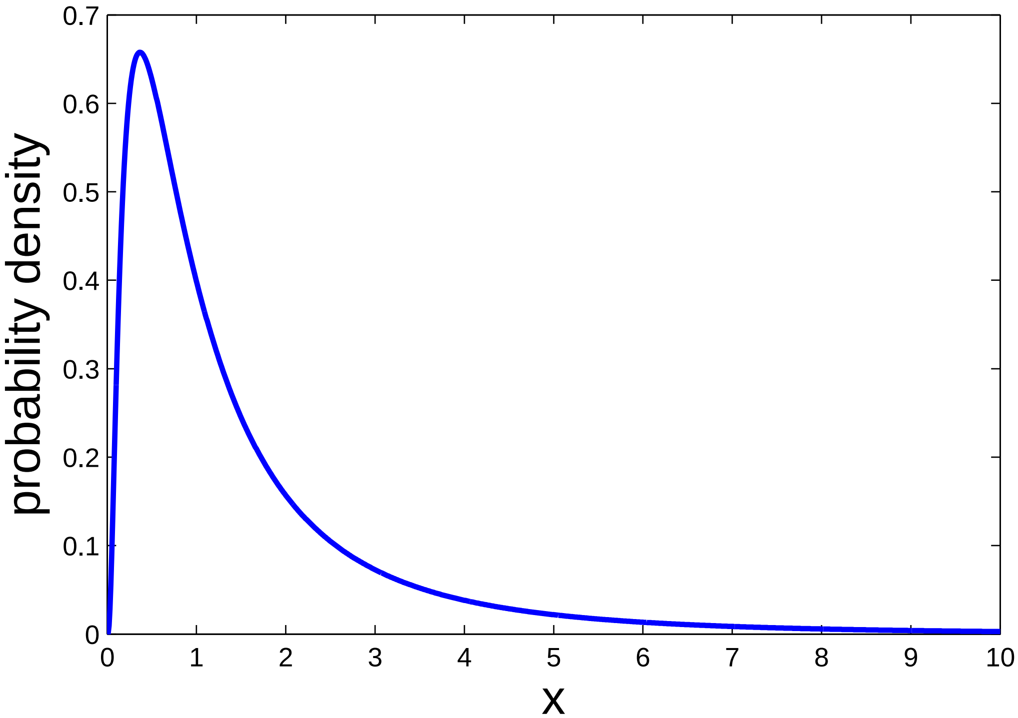

Figure 5.2: Lognormal distribution probability density function, with $\alpha = 0$ and $\beta = 1$.

distribution function for the lognormal distribution may be written in terms of the *error function*:

$$F(x) = \frac{1}{2} + \frac{1}{2}\,\text{erf}\left(\frac{\ln x - \alpha}{\beta\sqrt{2}}\right) \tag{5.45}$$

In `MATLAB` the lognormal cumulative distribution function is given by the command `logncdf`. There is no equivalent command in Microsoft `EXCEL`.

5.3.3 Laplace Distribution

Parameters

The Laplace distribution is a peaky (leptokurtic), symmetric distribution defined by two independent parameters:

μ the mean, which is a measure of location; and

$\alpha > 0$, which is a measure of spread.

All other parametric statistics of the Laplace distribution can be given in terms of μ and α. See Table 5.3.

Table 5.3: Laplace distribution parameters and notation.

mean	μ
median	μ
mode	μ
standard deviation	$\alpha\sqrt{2}$
variance	$2\alpha^2$
skewness	0
kurtosis	6
lower quartile	$\mu - 0.6931\alpha$
upper quartile	$\mu + 0.6931\alpha$
inter-quartile range	1.3863α

Probability Density Function

The probability density function for the Laplace distribution is given by [10, 33]

$$f(x) = \frac{1}{2\alpha} \exp\left(-\frac{|x-\mu|}{\alpha}\right) \tag{5.46}$$

for $-\infty < x < +\infty$. Equation (5.29) is symmetric about $x = \mu$. See Figure 5.3.

The Laplace distribution probability density function is not defined in MATLAB or in Microsoft EXCEL, although it can be constructed.

Cumulative Distribution Function

The cumulative distribution function for the Laplace distribution is given by

$$F(x) = \begin{cases} \frac{1}{2} \exp\left(\frac{x-\mu}{\alpha}\right) & x < \mu \\ 1 - \frac{1}{2} \exp\left(\frac{\mu-x}{\alpha}\right) & x \geqslant \mu \end{cases} \tag{5.47}$$

5.3.4 Generalized Normal Distribution

Parameters

Unlike the Normal, Lognormal, and Laplace distributions with two parameters, the Generalized Normal distribution has three independent parameters [22].

μ the mean, which is a measure of location;

$\alpha > 0$, which is a measure of spread; and,

$\beta > 0$, which is a measure of shape.

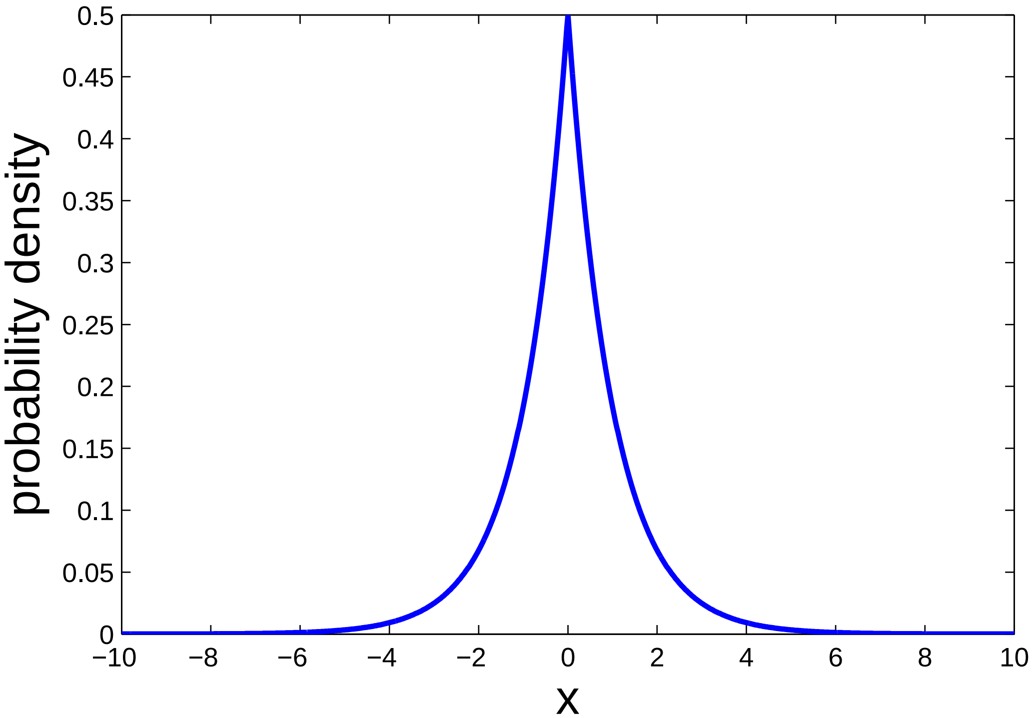

Figure 5.3: Laplace distribution probability density function, with $\mu = 0$ and $\alpha = 1$.

All other parametric statistics of the Normal distribution can be given in terms of μ, α, and β. See Table 5.4.

The Generalized Normal distribution is also known as the *Exponential Power distribution*. The Generalized Normal distribution is a remarkably flexible symmetric distribution. The Generalized Normal distribution includes the Normal distribution as a special case when $\beta = 2$; the resulting variance is $\frac{\alpha^2}{2}$. When $\beta > 2$ the Generalized Normal distribution is flat-topped (platykurtic). In the limit as $\beta \to \infty$, the Generalized Normal distribution becomes a Uniform (Rectangular) distribution. when $\beta < 2$ the Generalized Normal distribution becomes peaky (leptokurtic). For example, when $\beta = 1$ the Generalized Normal distribution is equivalent to the Laplace distribution.

Probability Density Function

The probability density function for the Normal distribution is given by [30]

$$f(x) = \frac{\beta}{2\alpha\Gamma(1/\beta)} \exp\left(-(|x - \mu|/\alpha)^{\beta}\right) \tag{5.48}$$

for $-\infty < x < +\infty$, where Γ denotes the *gamma function*. Equation (5.48) is symmetric about $x = \mu$, but the shape depends on the β parameter; see Figure 5.4.

The Generalized Normal distribution probability density function is not defined in `MATLAB` or in Microsoft `EXCEL`.

Table 5.4: Generalized Normal distribution parameters and notation.

mean	μ
median	μ
mode	μ
standard deviation	$\alpha\sqrt{\frac{\Gamma(3/\beta)}{\Gamma(1/\beta)}}$
variance	$\frac{\alpha^2\Gamma(3/\beta)}{\Gamma(1/\beta)}$
skewness	0
kurtosis	$\frac{\Gamma(5/\beta)\Gamma(1/\beta)}{\Gamma(3/\beta)^2}$
lower quartile	too complex to write explicitly
upper quartile	too complex to write explicitly
inter-quartile range	too complex to write explicitly

Cumulative Distribution Function

The cumulative distribution function for the Generalized Normal distribution can be written in terms of the *gamma function* and the *lower incomplete gamma function*

$$F(x) = \begin{cases} \frac{\gamma\left(1/\beta, \left(\frac{\mu-x}{\alpha}\right)^\beta\right)}{2\Gamma(1/\beta)} & x \leqslant \mu \\ 1 - \frac{\gamma\left(1/\beta, \left(\frac{x-\mu}{\alpha}\right)^\beta\right)}{2\Gamma(1/\beta)} & x > \mu \end{cases} \tag{5.49}$$

where Γ denotes the *gamma function*, and γ denotes the *lower incomplete gamma function*.

5.4 Outlier Analysis

5.4.1 Background

At the heart of many statistical tests is a simple range check: if a datum or a statistic falls outside of a pre-specified range then the datum or associated data are deemed outliers. For the purposes of this project, such outliers are identified as suspect data.

While the test itself is very simple, defining the appropriate range proves to be more involved. If the chosen range is too small, an excessive amount of good data will be lost under the guise of being bad (these are called "false positives"). Often, it is the extremes that are of interest, so losing too many of them is unacceptable. Conversely, if the range is too big, the test will be useless as it will see bad data as just more extreme, but acceptable, data (these are called "false-negatives").

Often, the standard deviation times a multiplier is used to define an acceptable range:

$$\mu - \alpha\sigma < X < \mu + \alpha\sigma \tag{5.50}$$

where

- X is the datum or statistics under consideration,

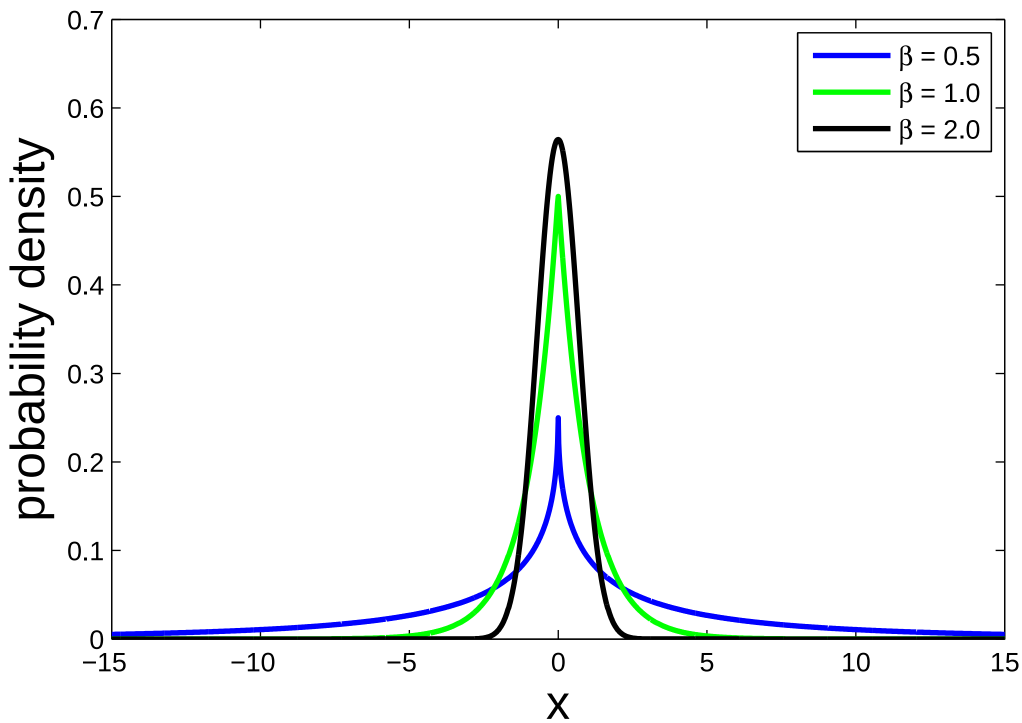

Figure 5.4: Generalized Normal probability density function, with $\mu = 0$ and $\alpha = 1$.

- μ is the mean of X,
- σ is the standard deviation of X, and
- α is a predefined multiplier.

An example of such a range would be the classical statistical result for the Normal Distribution that says: "the mean plus or minus two standard deviations captures approximately 95% of the population".

The concept presented by (5.50) is sound, but there are three practical difficulties:

1. Not all populations follow a Normal Distribution. In fact, we are never certain about the underlying distribution.
2. Not all populations are symmetric. In the case of an asymmetric distribution, like a Lognormal Distribution, the acceptable range should be asymmetric about the mean.
3. If the acceptable range is based upon the mean and the standard deviation, as in (5.50), then errors in the estimated mean and standard deviation yield errors in the range.

The solutions embraced in this project to address these three practical difficulties are equally practical.

1. An appropriate distributional form for each of the tests must be inferred from the considered data. Fortunately, for most tests there is a great deal of data available. It was found that one of the four theoretical distributions introduced in Section 5.3 (Normal, Lognormal, Laplace, or Generalized Normal) was adequate for each of the tests performed.

2. Following common practice, it was decided to define the acceptable range so that the probability of a false positive is evenly split between the two tails. Thus, if X is a good datum (not an outlier) then

$$\boldsymbol{Pr}(\mathrm{X} < \text{lower bound}) = \boldsymbol{Pr}(\mathrm{X} > \text{upper bound}) = \frac{\text{probability of a false positive}}{2} \tag{5.51}$$

   This simple rule will generate ranges that are asymmetric about the mean when the selected statistical distribution is asymmetric.

3. The standard estimates for the mean (5.2) and standard deviation (5.6) are sensitive to outliers in the data (see, for example, [9, Chapter 8]). This creates a *catch-22*: the outliers must be eliminated before estimating the mean and standard deviation, but the estimates of the mean and standard deviation are necessary to apply (5.50). To "inoculate" the computed acceptable ranges from the bad data, the median is used as the estimate of location and the upper and lower quartile spreads are used to estimate the width of distribution.

5.4.2 False Positives

Whenever a range test is performed, there will always be some errors. It is important to balance the number of false-positives and false-negatives. In this project the acceptable probability of a false positive[2] was selected to be 0.00001; this is 1 in 100,000. Thus, if all 470+ million data were, in fact, good, the statistical tests would still flag approximately

$$0.00001 \times 471{,}178{,}324 \approx 4{,}711 \tag{5.52}$$

as bad. This value of 1 in 100,000 was selected because it is a power of ten that gives around 4 false negatives per sensor. While this selection is arbitrary, it is not capricious. Numerous alternatives were considered, and 1 in 100,000 was deemed acceptable for the myriad tests.

5.4.3 Acceptable Range

General Result

For notational brevity, let p represent the selected probability of a false positive. The acceptable range is given by the general relationships

$$\boldsymbol{Pr}(\mathrm{X} < \text{lower bound}) = \mathrm{p}/2 \tag{5.53}$$

$$\boldsymbol{Pr}(\mathrm{X} > \text{upper bound}) = \mathrm{p}/2 \tag{5.54}$$

[2]This parameter value is set in the function `CreateInitialization.m` using the manifest constant `PROBABILITY_OF_FALSE_POSITIVE`.

Let F represent the cumulative distribution function of the selected distribution (i.e. Normal, Lognormal, Laplace, or Generalized Normal), and F^{-1} its inverse, then we can solve (5.57) for the lower and upper bounds as

$$\text{lower bound} = F^{-1}(p) \tag{5.55}$$

$$\text{upper bound} = F^{-1}(1-p) \tag{5.56}$$

Normal Distribution

Let

$$\zeta = \frac{F^{-1}(1-p)}{F^{-1}(0.75)} \tag{5.57}$$

where p represents the selected probability of a false positive, and F^{-1} is the inverse of the Normal cumulative distribution function (5.30). With $p = 1/100,000$, (5.57) yields $\zeta \approx 6.3231$. The acceptable bounds are then computed as

$$\text{lower bound} = \tilde{x} + \zeta\,[x_{lq} - \tilde{x}] \tag{5.58}$$

$$\text{upper bound} = \tilde{x} + \zeta\,[x_{uq} - \tilde{x}] \tag{5.59}$$

where $\tilde{x}$, x_{lq}, and x_{uq} are computed using (5.13), (5.16), and (5.19). In MATLAB these computation takes to form

```
zeta = norminv(1-p)/norminv(0.75);

lower_bound = median + zeta*(lower_quartile - median);
upper_bound = median + zeta*(upper_quartile - median);
```

Lognormal Distribution

Again, let

$$\zeta = \frac{F^{-1}(1-p)}{F^{-1}(0.75)} \tag{5.60}$$

where p represents the selected probability of a false positive, and F^{-1} is the inverse of the Normal cumulative distribution function (5.30). With $p = 1/100,000$, (5.60) yields $\zeta \approx 6.3231$. The acceptable bounds are then computed as

$$\text{lower bound} = e^{\ln\tilde{x} + \zeta\left[\ln x_{lq} - \ln\tilde{x}\right]} \tag{5.61}$$

$$\text{upper bound} = e^{\ln\tilde{x} + \zeta[\ln x_{uq} - \ln\tilde{x}]} \tag{5.62}$$

where $\tilde{x}$, x_{lq}, and x_{uq} are computed using (5.13), (5.16), and (5.19). In MATLAB these computation takes to form

```
zeta = norminv(1-p)/norminv(0.75);

lower_bound = exp( log(median) +
    zeta*( log(lower_quartile) - log(median) ) );
upper_bound = exp( log(median) +
    zeta*( log(upper_quartile) - log(median) ) );
```

Laplace Distribution

The Laplace distribution is handled as a special case of the Generalized Normal distribution.

Generalized Normal Distribution

The Generalized Normal Distribution is not amenable to the relatively simple representation offered by the Normal and Lognormal distributions. Rather, it is necessary to apply (5.55) and (5.56) directly. In MATLAB these computations take the form

```
lowerbound = gnorminv(  p/2, mu, alpha, beta);
upperbound = gnorminv(1-p/2, mu, alpha, beta);
```

The routine `gnorminv` is not part of the standard MATLAB distribution, it is a function written specifically for this project. This function is included at in the `Flag_NeighborhoodOutliers.m` file.

Chapter 6

Testing and Flags

6.1 Who, What, Where, When, Why

There are many ways that data can exhibit erroneous, questionable, or exceptional behavior. The general categories of such behaviors are outlined in Chapter 3 of this report. Some erroneous data are obvious by simple visual inspection at the coarsest level. For example, temperature values of $-2000°C$ recorded for Sensor 1 of Cell #93 are false. Other questionable data may be identified by a meticulous inspection on a day-by-day, hour-by-hour, or datum-by-datum basis. For example, a $5°C$ jump in the temperature over a 15 minute sample interval is too large for a sensor at depth. More insidious exceptional data appear reasonable when viewed as isolated measurements or as a simple time series, but their exceptional nature is revealed by comparison with contemporaneous data from similar sensors.

With over four hundred and seventy million temperature records in the existing data base it is impossible to manually consider, and sift, and filter to the level required. Furthermore, many of the exceptional data cannot be distinguished by visual inspection; rather, it is necessary to make a computationally intense statistical comparison. In short, this process must be automated.

Fourteen categories of erroneous, questionable, or exceptional behavior have been identified. Each of these categories captures a diagnostic characteristic for the data and merits unique consideration, although there is some overlap between the categories. Each of these fourteen categories has an identification algorithm and computational engine. Most of the remainder of this chapter presents the associated details.

6.2 Flags and Flag Arrays

6.2.1 Motivation

Data that are identified as erroneous, questionable, or exceptional could be simply eliminated from the database. However, such a severe approach presupposes that the filtering is perfect and disallows all future considerations of the deleted data. Furthermore, even erroneous data may contain useful information – for an investigation into what went wrong, if nothing else.

Rather than eliminating data, it was decided to construct a set of parallel *flag files*. These flag files indicate which of the temperature records exhibit erroneous, questionable, or exceptional behavior, and which temperature records do not.

6.2.2 Binary Flags

A *bit* is a single **b**(inary) (dig)**it**. There are two, and only two, possible values for a single bit: 0 or 1. A 16-bit unsigned integer comprises sixteen bits

[`bbbb bbbb bbbb bbbb`]

where each bit, represented by a "b", is a 0 or a 1. Two bytes (16-bits) and four bytes (32-bits) are the standard storage format for integer data in modern computers. For example, in the C++ computer programming language, these are known as `short` and `int` [25]. MATLAB supports 8, 16, and 32 bit versions of signed and unsigned integer types: `int8`, `uint8`, `int16`, `uint16`, `int32`, and `uint32`.

As with standard decimal integers, place values are associated with the individual bits. In this case, however, the place values are powers of 2 and not powers of 10 (see Table 6.1). The bits are numbered right-to-left: that is, the right-most bit is the "ones" place.

Table 6.1: Places values for a 16-bit unsigned integer.

bit number	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
place value	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

For example, the following 16-bit unsigned integer may be written in binary form as

[0000 0000 1100 0101]

or equivalently, as a standard decimal number

$$
\begin{aligned}
&= 0 \times 2^{15} + 0 \times 2^{14} + 0 \times 2^{13} + 0 \times 2^{12} + 0 \times 2^{11} + 0 \times 2^{10} + 0 \times 2^{9} + 0 \times 2^{8} + \\
&\quad 1 \times 2^{7} + 1 \times 2^{6} + 0 \times 2^{5} + 0 \times 2^{4} + 0 \times 2^{3} + 1 \times 2^{2} + 0 \times 2^{1} + 1 \times 2^{0} \\
&= 128 + 64 + 4 + 1 \\
&= 197
\end{aligned}
\tag{6.1}
$$

Each temperature record is associated with a 16-bit unsigned integer flag. Each of the first fourteen bits of a flag is associated with an identified category of erroneous, questionable, or exceptional behavior. (Note that this leaves two, as yet, unused bits for future tests. By default, these unused bits are explicitly set to 0.) If a temperature record fails the first test, then the first bit of the associated flag is set to 1, otherwise the first bit is set to 0. If a temperature record fails the second test, then the second bit of the associated flag is set to 1, otherwise the second bit is set to 0. And so on for all of the fourteen tests. Computer programming languages include specific commands to manipulate individual bits simply and efficiently. In MATLAB, these commands include: `bitset`, `bitget`, `bitand`, `bitor`, `bitcmp`.

It is possible, and quite common, for a bad datum to fail multiple tests. When this occurs the associated flag has multiple bits set to 1. For example, if a temperature record is bad and needs flags 1, 3, 7 and 8 set, the 16 bits will be set as

$$[\,0000\ \ 0000\ \ 1100\ \ 0101\,]$$

in binary, which number will actually print out as its base-ten equivalent: 197.

If a temperature record passes all fourteen tests, then all of the bits of the associated flag are set to 0. Thus, flag = 0 indicates a good datum (i.e. no flags are set), while flag = anything else indicates a problem datum (i.e. at least one flag is set).

6.2.3 Flag Arrays

Every temperature record has an associated 16-bit unsigned integer flag. As described in Section 2.5, every cell has a `.mat` file. For example, `Cell022.mat` contains all of the data for Cell #22. In each of these `.mat` files there are three arrays: D, T, and F:

- D: (N × 1) array of dates using MATLAB's `datenum` format,
- T: (N × S) array of temperature data,
- F: (N × S) array of 16-bit unsigned integer flags.

where N is the total number of observations for the cell (the sum over all of the years) and S is the number of sensors in the cell. The F array is the "flag array".

6.3 Enumeration of Flags

To aid in the evaluation and reparation of erroneous, questionable, or exceptional data, each binary flag represents a specific problem with the observation. The flags numbers are fully enumerated in Table 6.2. A set of general properties for the flags are summarized in Table 6.3.

6.4 Missing Data

6.4.1 The Flag

The details of the missing-data flag are show in table 6.4.

6.4.2 Identifying Missing Data

The most basic type of "bad" data is when there is no data at all. Each sensor in a cell reports back data at the same time, which allows for only one date matrix per cell. However, when a sensor is taken off-line, or before it comes on-line, or stops reporting temperatures for any reason, something still needs to be returned for that time period. In such circumstances, the sensor records a zero.

In the raw ASCII source files, missing data was designated with a value of "0". This was an unfortunate choice since 0 is a valid temperature reading that is not uncommon in Minnesota.

From 1998 to the present, the temperature data is represented using four or more digits to the right of the decimal point. As such, it is highly unlikely that a temperature reading of exactly

Table 6.2: A complete enumeration of flag numbers.

Flag Number	Base 2 Value	Base 10 Value	Flag Name
1	0000 0000 0000 0001	1	`FLAG_MISSING_DATA`
2	0000 0000 0000 0010	2	`FLAG_NOT_YET_OPERATIONAL`
3	0000 0000 0000 0100	4	`FLAG_DEACTIVATED`
4	0000 0000 0000 1000	8	`FLAG_TOO_SPARSE_DAY`
5	0000 0000 0001 0000	16	`FLAG_OUT_OF_RANGE`
6	0000 0000 0010 0000	32	`FLAG_NEIGHBORHOOD_OUTLIERS`
7	0000 0000 0100 0000	64	`FLAG_LAG_ONE_OUTLIERS`
8	0000 0000 1000 0000	128	`FLAG_POINT_EXTREMES`
9	0000 0001 0000 0000	256	`FLAG_DAILY_RANGE`
10	0000 0010 0000 0000	512	`FLAG_DAILY_EXTREMES`
11	0000 0100 0000 0000	1024	`FLAG_INTERMITTENT_DATA`
12	0000 1000 0000 0000	2048	`FLAG_INCONSISTENT_DAY`
13	0001 0000 0000 0000	4096	`FLAG_INCONSISTENT_WEEK`
14	0010 0000 0000 0000	8192	`FLAG_INCONSISTENT_MONTH`
15	0100 0000 0000 0000	16384	NOT USED
16	1000 0000 0000 0000	32768	NOT USED

Table 6.3: A list of various general flag properties.

Property	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Missing data type flag	✓	✓	✓	✓										
Time series based flag					✓	✓	✓							
Subset based flag								✓	✓	✓				
Consistency flag											✓	✓	✓	✓
One or more predecessor flags				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Uses `SubsetRanges.mat`								✓						
Uses `DailyRanges.mat`									✓	✓				
Must be run in batch mode								✓	✓	✓				
Uses Normal model					✓			✓		✓				
Uses Lognormal model									✓					
Uses General. Normal model						✓	✓							
High Computational Effort	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓
Very High Computational Effort						✓	✓							
Computational Effort Rank	8	14	13	9	3	2	1	7	6	5	4	10	12	11

Table 6.4: The missing data flag.

name	`FLAG_MISSING_DATA`
bit number	1
binary mask	0000 0000 0000 0001
decimal value	1

0.0000 is collected. (No recorded values of 0.0000 were found in the data analyzed.) Thus, after 1998, a value of exactly 0 in the raw data file can be reliably designated as a missing value.

Prior to 1998, the temperature data is represented using at most two digits to right of the decimal, and sometimes even these are truncated. As such, some correct temperature measurements are encoded as 0, which is indistinguishable from a missing value.

The reliable identification of missing data prior to 1998 requires some analysis, and a few assumptions. For example, if a sequence of temperature measurements read

$$\{0.6, 0.4, 0.2, 0, -0.2, -0.4...\} \tag{6.2}$$

it is reasonable to believe that the 0 reading is a correct measurement and not a missing data value. On the other hand, if a sequence of temperature measurements read

$$\{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...\} \tag{6.3}$$

it is highly unlikely that the temperature would stay exactly equal to 0 for a long period of time, so these should be interpreted as a sequence of missing data.

A recorded temperature value of 0 is flagged as missing if any one of the following five conditions is true:

- year $\geqslant$ 1998;
- it is the first measurement of the first year;
- the previous measurement is missing;
- there are more than `MAX_RUN_AT_0` consecutive 0's; or
- the previous measurement differs by more than `MAX_DELTA_T_AT_0`;

where `MAX_RUN_AT_0` and `MAX_DELTA_T_AT_0` are manifest constants defined in the initialization function[1]. The current values for these constants are

$$\texttt{MAX\_RUN\_AT\_0} = 24 \tag{6.4}$$

$$\texttt{MAXIMUM\_DELTA\_T\_AT\_0} = 2 \tag{6.5}$$

6.4.3 `Flag_MissingData`

The details of the missing-data function are show in table 6.5.

Purpose

Identify and flag missing data (temperature fields) within the data records present in all cells and all sensors.

[1] See `CreateInitialization.m`

Table 6.5: The flag missing data function.

file	`Flag_MissingData.m`
usage	`[] = Flag_MissingData()`
arguments	NONE
returns	NONE

Preconditions

- An up-to-date ”Initialization.mat” file must exist in the current working directory.
- The binary source files (i.e. `Cellxxx.mat`) must exist in the source folder identified in the initialization file.

6.5 Not Yet Operational

6.5.1 The Flag

The details of the not-yet-operational flag are show in table 6.6.

Table 6.6: The not yet operational flag.

name	`FLAG_NOT_YET_OPERATIONAL`
bit number	2
binary mask	0000 0000 0000 0010
decimal value	2

6.5.2 Identifying Not Yet Operational Data

While the earliest recorded MnROAD temperature in the considered data set is from January 1996, some of the sensors did not begin recording data until much later. Because of this, the first few years of a sensor's data record are comprised of only “missing data”. While this data should be excluded, it would not be accurate to merely bundle these data into the same group as erroneous missing data. Consequently, the *Not Yet Operational* flag was created to highlight these circumstances.

The process in which the function flags the not yet operational readings is a fairly simple one. It starts at a sensor's very first reading and continues through time until it finds a good, numerical datum. Every reading before this point is flagged as not yet operational. See, for example, Figure 3.1.

It is anticipated that, under most circumstances, all data flagged as “not yet operational” will also be flagged as “missing”. This flag is, nonetheless, useful since it gives additional information: an explanation for the associated missing data.

6.5.3 `Flag_NotYetOperational`

he details of the not yet operational function are show in table 6.7.

Table 6.7: The flag missing data function.

file	`Flag_NotYetOperational.m`
usage	`[] = Flag_NotYetOperational()`
arguments	NONE
returns	NONE

Purpose

For each sensor, flag all data until the first real reading.

Preconditions

- An up-to-date ”Initialization.mat” file must exist in the current working directory.
- The binary source files (i.e. `Cellxxx.mat`) must exist in the source folder identified in the initialization file.
- The following flags have already been correctly applied:
  - `FLAG_MISSING_DATA`

6.6 Deactivated

6.6.1 The Flag

The details of the deactivated flag are show in table 6.8.

Table 6.8: The deactivated flag.

name	`FLAG_DEACTIVATED`
bit number	3
binary mask	0000 0000 0000 0100
decimal value	4

6.6.2 Identifying Deactivated Data

Similar to the *Not Yet Operational* function, the Deactivated function identifies and flags data in a sensor which had stopped taking temperature readings indefinitely.

The function begins at the end of the sensor data and works backwards through time until it finds the first good, numerical datum. All readings beyond that point are flagged as Deactivated. See, for example, Figure 3.2.

It is anticipated that, under most circumstances, all data flagged as "deactivated" will also be flagged as "missing". This flag is, nonetheless, useful since it gives additional information: an explanation for the associated missing data.

6.6.3 `Flag_Deactivated`

The details of the deactivated flag are show in table 6.9.

Table 6.9: The flag deactivated data function.

file	`Flag_Deactivated.m`
usage	`[] = Flag_Deactivated()`
arguments	NONE
returns	NONE

Purpose

For each sensor, flag all data after the last real reading.

Preconditions

- An up-to-date "Initialization.mat" file must exist in the current working directory.
- The binary source files (i.e. `Cellxxx.mat`) must exist in the source folder identified in the initialization file.
- The following flags have already been correctly applied:
  - `FLAG_MISSING_DATA`

6.7 Too Sparse Day

6.7.1 The Flag

The details of the too-sparse-day flag are show in table 6.10.

Table 6.10: The too sparse day flag.

name	FLAG_TOO_SPARSE_DAY
bit number	4
binary mask	0000 0000 0000 1000
decimal value	8

6.7.2 Identifying Too Sparse Day Data

In almost all cases, data is collected from active sensors every 15 minutes. Thus, in a normal day there are $4 \cdot 24 = 96$ temperature measurements collected from each sensor. If, in any day, a sensor collects significantly fewer than 96 temperature measurements then all of the data from that day is suspect. These data are not necessarily bad, but they are tainted by proximity to bad or missing data. See, for example, Figure 3.3 and Figure 3.4.

The data record for every day, and for each sensor is considered. If there are fewer than a predetermined number, `MINIMUM_DATA_PER_DAY`, in any day for any sensor, all data from that day and from that sensor are flagged as "too sparse day". The manifest constant `MINIMUM_DATA_PER_DAY` is defined in the initialization[2] function. The current value for this constant is

$$\texttt{MINIMUM\_DATA\_PER\_DAY} = 48 \tag{6.6}$$

For the purposes of this assessment, a day runs from midnight to midnight.

6.7.3 `Flag_TooSparseDay`

The details of the too-sparse-day flag are show in table 6.11.

Table 6.11: The flag too sparse day function.

file	`Flag_TooSparseDay.m`
usage	`[] = Flag_TooSparseDay()`
arguments	NONE
returns	NONE

Purpose

For each sensor, all records from days with fewer than a minimum number of data are flagged.

Preconditions

- An up-to-date "Initialization.mat" file must exist in the current working directory.

[2]See `CreateInitialization.m`

- The binary source files (i.e. `Cellxxx.mat`) must exist in the source folder identified in the initialization file.
- The following flags have already been correctly applied:
  - FLAG_MISSING_DATA
  - FLAG_NOT_YET_OPERATIONAL
  - FLAG_DEACTIVATED

6.8 Out of Range

6.8.1 The Flag

The details of the out-of-range flag are show in table 6.12.

Table 6.12: The out of range flag.

name	FLAG_OUT_OF_RANGE
bit number	5
binary mask	0000 0000 0001 0000
decimal value	16

6.8.2 Identifying Out of Range Data

The analysis for this flag is carried out on a sensor-by-sensor basis. When identifying outliers for each sensor, only data from that specific sensor's time series are used.

Spectral Analysis

Visual inspection of the data records for temperatures from properly functioning, active sensors shows seasonal (annual) and diurnal (daily) periodic behavior. These anticipated frequencies are confirmed[3] by the amplitude spectrum [3]: see, for example, Figure 6.1. Statistically significant peaks occur at a very low frequency, at 1 cycle per day, 2 cycles per day, 3 cycles per day, and at 4 cycles per day [23]. Re-plotting the amplitude spectrum with wavelength on the abscissa, highlights the low frequency component and shows that it has a wavelength of one year (see Figure 6.2). The amplitude spectrum shown in these two plots is for the complete data record (1996-2007) from Cell #1, Sensor 1, which is quite shallow (depth = 0.1 [ft]).

Similar periodic behavior occurs even from relatively deep sensors. Figures 6.3 and 6.4 show comparable plots for Cell #10, Sensor 11. This sensor is in the clay subgrade at 2.4293 m (7.97 ft)

[3]Spectral analysis of this sort is greatly simplified by the built-in tools in MATLAB, including the fast Fourier transform function `fft`. More advanced tools are available in the "Signal Processing Toolbox", but these did not add significant insight within the context of this report.

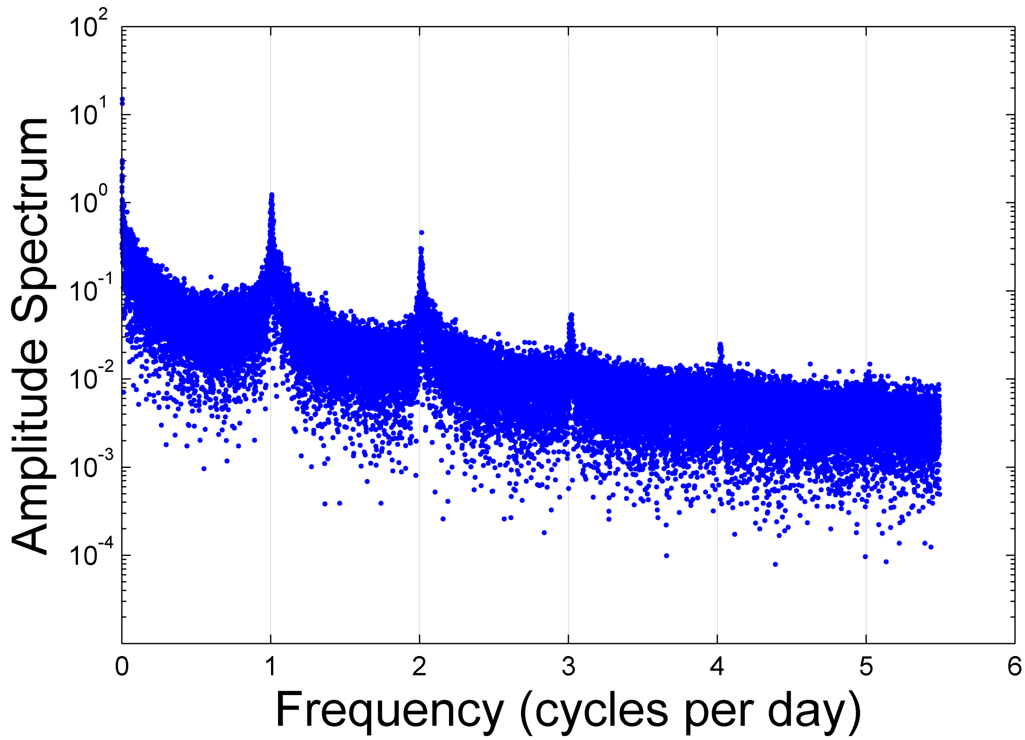

Figure 6.1: Single-sided amplitude spectrum of temperatures for the complete data record (1996-2007) from Cell #1, Sensor 1. This is an HMA cell on the Mainline, and the sensor is 0.0305 $\mathtt{m}$ $(0.1\ \mathtt{ft})$ below the surface.

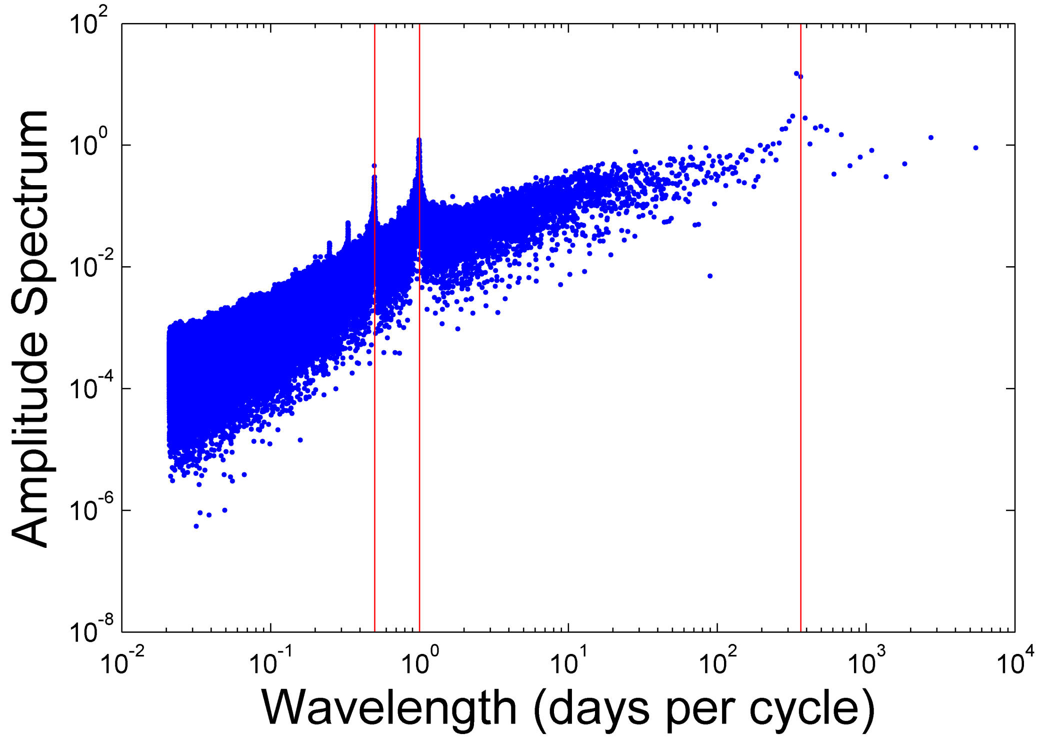

Figure 6.2: Single-sided amplitude spectrum of temperatures for the complete data record (1996-2007) from Cell #1, Sensor 1. This is an HMA cell on the Mainline, and the sensor is 0.0305 m (0.1 ft) below the surface. The vertical red lines show wavelengths of $\frac{1}{2}$ day, 1 day, and 1 year.

below the surface. At depth, the spikes in the amplitude spectrum at 3 and 4 cycles per day disappear, and the spike at 2 cycles per day is attenuated. This behavior is consistent across all of the properly functioning, active sensors.

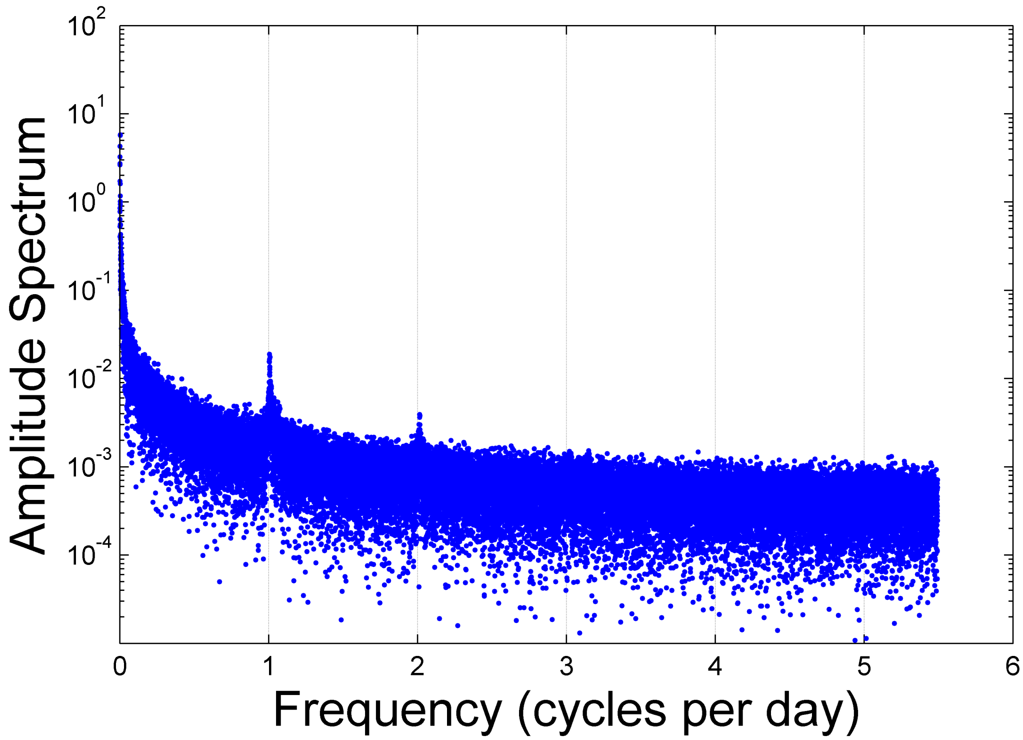

Figure 6.3: Single-sided amplitude spectrum of temperatures from Cell #10, Sensor 11. This is a PCC cell on the Mainline, and the sensor is 2.4293 m (7.97 ft) below the surface in the clay subgrade.

Computing the Residuals

The periodic seasonal (annual) and diurnal (daily) temperature variations are removed using a least-squares fit to the following simple periodic model:

$$\begin{aligned} T(t) = & T_0 + \\ & a_1 \cos(2\pi t/365.25) + b_1 \sin(2\pi t/365.25) + \\ & a_2 \cos(2\pi t) + b_2 \sin(2\pi t) + \\ & r(t) \end{aligned} \tag{6.7}$$

where

- t is the time in days,
- $T(t)$ is the temperature at time t,
- T_0 is the fitted mean temperature for the entire time series,

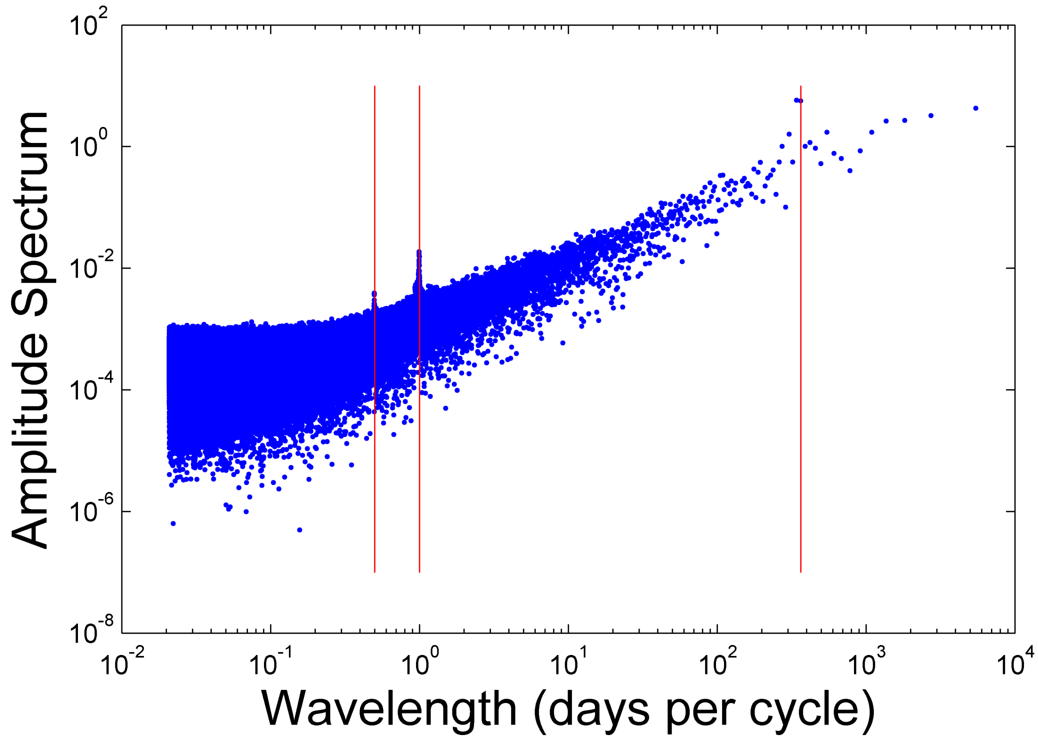

Figure 6.4: Single-sided amplitude spectrum of temperatures from Cell #10, Sensor 11. This is a PCC cell on the Mainline, and the sensor is 2.4293 m (7.97 ft) below the surface in the clay subgrade. The vertical red lines show wavelengths of $\frac{1}{2}$ day, 1 day and 1 year.

- (a_1, b_1) are the fitted coefficients for the annual variation,
- (a_2, b_2) are the fitted coefficients for the diurnal variation, and
- $r(t)$ is the residual at time t.

Only the annual and daily periodic components are removed; the components identified at wavelengths of $\frac{1}{2}$, $\frac{1}{3}$, and $\frac{1}{4}$, days are not explicitly accounted for. There are three reasons for this decision:

1. The annual and the daily components are dominant.
2. The higher frequency components disappear at depth, and the simplicity of a single model for all sensors is attractive.
3. The higher frequency components offered no additional probative value when applied to the numerous shallow sensors.

Outlier Identification

For all properly functioning, active sensors the distribution of the residuals computed using 6.7 is well-modeled by a Normal distribution (see Section 5.3.1).

The outlier identification, outlined in Section 5.4, is applied to the residuals.

- The upper and lower quartile spread for the residuals are defined by

$$\text{upper quartile spread} = \text{75th percentile} - \text{50th percentile} \tag{6.8}$$
$$\text{lower quartile spread} = \text{50th percentile} - \text{25th percentile} \tag{6.9}$$

- The upper and lower bounds for the normal residuals are defined as

$$\text{upper bound} = \text{median} + \zeta \cdot \text{upper quartile spread} \tag{6.10}$$
$$\text{lower bound} = \text{median} - \zeta \cdot \text{lower quartile spread} \tag{6.11}$$

  where ζ is defined such that

$$\boldsymbol{Pr}(r(t) < \text{lower bound}) = \texttt{PROBABILITY\_OF\_FALSE\_POSITIVE}/2 \tag{6.12}$$
$$\boldsymbol{Pr}(r(t) > \text{upper bound}) = \texttt{PROBABILITY\_OF\_FALSE\_POSITIVE}/2 \tag{6.13}$$

  where `PROBABILITY_OF_FALSE_POSITIVE` is a manifest constant defined in the initialization[4] function. The current value for this constant is:

$$\texttt{PROBABILITY\_OF\_FALSE\_POSITIVE} = 1/100000 \tag{6.14}$$

- Any residuals exceeding the upper or lower bounds are identified as outliers and the associated data are flagged as out-of-range.

Cell #6, Sensor 7 is an excellent demonstration of the discerning power of this test (see Figure 6.5). The test identifies the anomalous data in the summer of 2006, even though all of the data is within the range of normal annual variation.

[4]See `CreateInitialization.m`

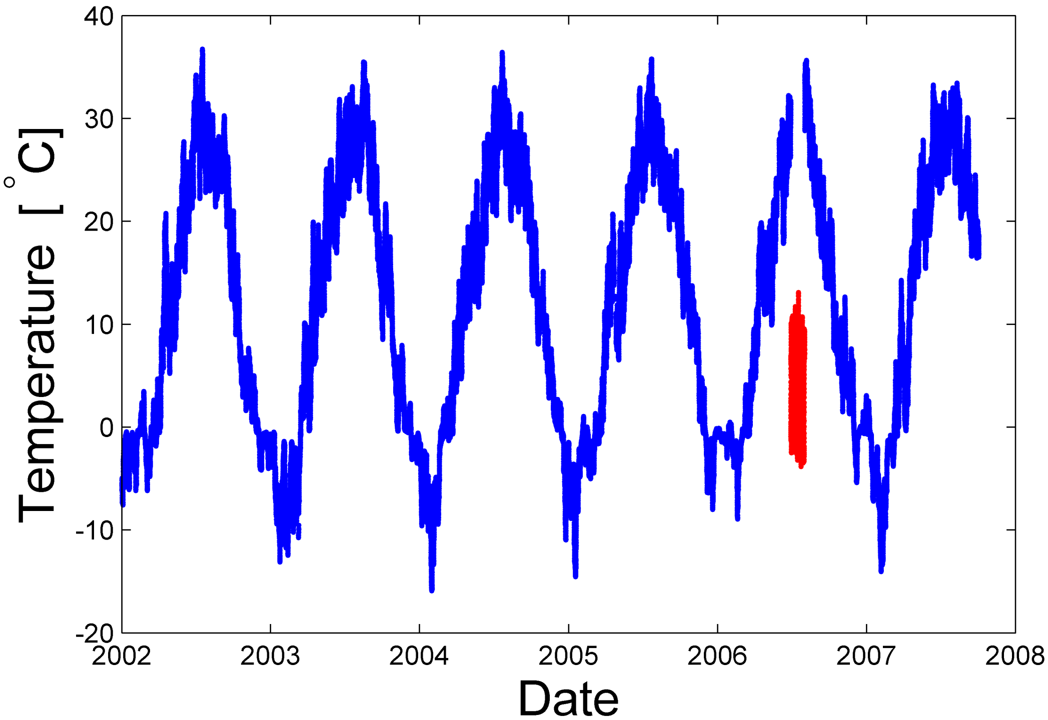

Figure 6.5: Temperature data from Cell #6, Sensor 5 during 2002 through 2007. The red markers show the data that is flagged by the "out of range" test. This test correctly identifies the anomalous data in the summer of 2006, even though all of the data is within the range of normal annual variation.

6.8.3 `Flag_OutOfRange`

The details of the out-of-range flag are show in table 6.13.

Table 6.13: The flag out of range data function.

file	`Flag_OutOfRange.m`
usage	`[] = Flag_OutOfRange()`
arguments	NONE
returns	NONE

Purpose

Identify and flag outliers on a sensor-by-sensor basis after removing the annual and diurnal periodic variation.

Preconditions

- An up-to-date "Initialization.mat" file must exist in the current working directory.
- The binary source files (i.e. `Cellxxx.mat`) must exist in the source folder identified in the initialization file.
- The following flags have already been correctly applied:
  - `FLAG_MISSING_DATA`
  - `FLAG_NOT_YET_OPERATIONAL`
  - `FLAG_DEACTIVATED`
  - `FLAG_TOO_SPARSE_DAY`

6.9 Neighborhood Outliers

6.9.1 The Flag

The details of the neighborhood-outliers flag are show in table 6.14.

Table 6.14: The neighborhood outliers flag.

name	`FLAG_NEIGHBORHOOD_OUTLIERS`
bit number	6
binary mask	0000 0000 0010 0000
decimal value	32

6.9.2 Identifying Neighborhood Outlier Data

The analysis for this flag is carried out on a sensor-by-sensor basis. When identifying outliers for each sensor, only data from that specific sensor's time series are used. Furthermore, only data that are NOT flagged with one of the following is considered in this analysis:

- `FLAG_MISSING_DATA`
- `FLAG_NOT_YET_OPERATIONAL`
- `FLAG_DEACTIVATED`
- `FLAG_TOO_SPARSE_DAY`
- `FLAG_OUT_OF_RANGE`

This test is based upon a temporal moving average of the recorded temperature data. The width of the moving neighborhood is given by the manifest constant `NEIGHBORHOOD_WIDTH`, which is defined in the initialization[5] function. The current value for this constant is:

$$\texttt{NEIGHBORHOOD\_WIDTH} = 3.0/24 \text{ [days]} \tag{6.15}$$

that is, 3 hours.

Consider the complete stream of n data from a sensor

$$\{(d_1, T_1), (d_2, T_2), \ldots, (d_n, T_n)\} \tag{6.16}$$

The i'th observation is an ordered pair (d_i, T_i) where d_i is the time that the measurement was taken (in [days]) and T_i is the measured temperature. We assume that the data pairs are sorted by increasing time.

The *moving neighborhood* for i'th observation is the set of all observations from the sensor of interest taken within the time interval

$$d_i \pm \texttt{NEIGHBORHOOD\_WIDTH}/2 \tag{6.17}$$

The *moving average* for the i'th observation, MA_i, is the arithmetic average of all temperature measurements in the moving neighborhood, NOT including the i'th measurement. Formally, let the set of indices J_i be defined by

$$J_i = \{j \text{ such that } j \neq i \text{ and } |d_j - d_i| \leqslant \texttt{NEIGHBORHOOD\_WIDTH}/2\} \tag{6.18}$$

then

$$MA_i = \frac{\sum_{j \in J} T_j}{\sum_{j \in J} 1} \tag{6.19}$$

The residual (moving average prediction error), r_i, is computed as

$$r_i = T_i - MA_i \tag{6.20}$$

[5]See `CreateInitialization.m`

Outlier identification is then applied to the residuals. The upper and lower bounds for the residuals are defined as presented in Section 5.4.3

$$\boldsymbol{Pr}(\mathtt{r_i} < \text{lower bound}) = \mathtt{PROBABILITY\_OF\_FALSE\_POSITIVE/2} \tag{6.21}$$
$$\boldsymbol{Pr}(\mathtt{r_i} > \text{upper bound}) = \mathtt{PROBABILITY\_OF\_FALSE\_POSITIVE/2} \tag{6.22}$$

Any residuals exceeding the upper or lower bounds are identified as outliers and the associated data are flagged as neighborhood outliers. The upper and lower bounds are computed based upon a fitted Generalized Normal Distribution for the residuals from the current sensor only.

6.9.3 `Flag_NeighborhoodOutliers`

The details of the neighborhood-outliers flag are show in table 6.15.

Table 6.15: The flag neighborhood outliers function.

file	`Flag_NeighborhoodOutliers.m`
usage	`[] = Flag_NeighborhoodOutliers()`
arguments	NONE
returns	NONE

Purpose

Identify and flag outliers on a sensor-by-sensor basis using a local neighborhood (temporal moving average) fit.

Preconditions

- An up-to-date "Initialization.mat" file must exist in the current working directory.
- The binary source files (i.e. `Cellxxx.mat`) must exist in the source folder identified in the initialization file.
- The following flags have already been correctly applied:
  - `FLAG_MISSING_DATA`
  - `FLAG_NOT_YET_OPERATIONAL`
  - `FLAG_DEACTIVATED`
  - `FLAG_TOO_SPARSE_DAY`
  - `FLAG_OUT_OF_RANGE`

6.10 Lag One Outliers

6.10.1 The Flag

The details of the lag-one-outliers flag are show in table 6.16.

Table 6.16: The lag one outliers flag.

name	`FLAG_LAG_ONE_OUTLIERS`
bit number	7
binary mask	0000 0000 0100 0000
decimal value	64

6.10.2 Identifying Lag One Outlier Data

The flagging of lag one values is fairly simple. As lag-one values are sporadic, it is difficult to compare values against a pattern. Consequently, a basic range-error is the most attractive workable option.

As lag-one is an exceptionally complicated idea, its explanation has been separated into a separate section (see Chapter 7). Once the underlying idea is understood, the actual flagging of lag-one values is fairly simple.

In order to facilitate the flagging of lag-one outliers, the grouping process needs to be further broken down into "subsets". Basically, subsets are the combination of all cell-wide groups into 54 giant groups (combining sensors of the same type, depth group, and location) for all of the data. If

$$T(s,t) > \tilde{G}(t) + \alpha \cdot (G(t)_{[75]} - \tilde{G}(t)) \tag{6.23}$$

or

$$T(s,t) < \tilde{G}(t) - \alpha \cdot (\tilde{G}(t) - G(t)_{[25]}) \tag{6.24}$$

where:

- $\tilde{G}$ is the median value of all lag-one values in subset 'G', at time t, and
- α is a predefined multiplier,

then the datum is flagged.

The lag-one flag is very important as it finds errors that pattern analysis flags generally do not. This having been said, however, it has a few weaknesses - primarily, a datum without an immediately preceding point cannot get checked for a lag-one value. Furthermore, large jumps in temperature are extremely important in analysis, so the outlier cutoff points have to be made reasonably tolerant as to not accidentally flag these important values. The occasional hole in the lag-one flagging process is an important reason for the intermittent and inconsistent data flags.

6.10.3 `Flag_LagOneOutliers`

The details of the lag-one-outliers function are show in table 6.17.

Table 6.17: The flag lag one outliers function.

file	Flag_LagOneOutliers.m
usage	[] = Flag_LagOneOutliers()
arguments	NONE
returns	NONE

Purpose

For each sensor, identify and flag outliers in the lag one differences.

Preconditions

- An up-to-date ”Initialization.mat” file must exist in the current working directory.
- The binary source files (i.e. Cellxxx.mat) must exist in the source folder identified in the initialization file.
- The following flags have already been correctly applied:
  - FLAG_MISSING_DATA
  - FLAG_NOT_YET_OPERATIONAL
  - FLAG_DEACTIVATED
  - FLAG_TOO_SPARSE_DAY
  - FLAG_OUT_OF_RANGE

6.11 Point Extremes

6.11.1 The Flag

The details of the point-extremes flag are show in table 6.18.

Table 6.18: The point extremes flag.

name	FLAG_POINT_EXTREMES
bit number	8
binary mask	0000 0000 1000 0000
decimal value	128

6.11.2 Identifying Point Extremes

In order to facilitate the flagging of point extremes, the grouping process needs to be further broken down into “subsets”. Basically, subsets are the combination all of cell-wide groups into 64 giant groups (combining sensors of the same type, depth group, and location) for all of the data. This allows for data comparison across cells as well as sensors. However, as inter-cell data does not necessarily correlate to the same time series, each group was transposed onto a uniform time series.

Under the assumption that sensors in the same group have similarly distributed temperature data, using the subsets to flag bad data becomes reasonably simple. For each point of time in the time series, a median temperature as well as the 75th and 25th percentiles can be calculated. If

$$T(s,t) > \tilde{G}(t) + \alpha \cdot (G(t)_{[75]} - \tilde{G}(t))$$
or
$$T(s,t) < \tilde{G}(t) - \alpha \cdot (\tilde{G}(t) - G(t)_{[25]})$$

where

- $\tilde{G}$ is the median value of all sensors in subset 'G' at time t, and
- α is a predefined multiplier,

then the datum is flagged.

During the creation of the subsets, it is assumed that between cells, sensors of similar pavement type and depth will have similar data. This, however, is not true. After performing a sign test on the sensor data, it is clear that they are biased. The sign test was performed by comparing each datum in a specific sensor to its respective subset median value. The number of sensor data greater than and less than the median are tallied for all of the sensor's data.

If the initial assumption of similar distributions was correct, one would expect about 50% of the data points to fall above the median and 50% to fall below. This would imply that the median value for the subset is close to the median value for the sensor. This did not happen. The output of a sample sign test for cell 53 is shown below. The two percentages do not add to 100% as a result of some data points being equal to the median value. Nonetheless, the percentages should be nearly equal, and they very clearly are not.
Furthermore, a plot of the sign test results for each sensor shows that the bias is not limited to a few cells. There are a few erroneous groups of sensors, but also very little grouping around 50%.

Because of this setback, the sensor bias needs to be counteracted before any sensor comparisons can take place. A sensor's subset range error is compared, term by term against the subset median at that point.

$$T(s,t) - \tilde{G}(t) = \Delta \tag{6.25}$$

where

T is the temperature matrix taking arguments s (sensor) and t (time)

$\tilde{G}$ is the median value of all sensors in subset 'G' at time t.

Δ is the sensor's deviation from the median

Table 6.19: The results of the Sign Test applied to sensors in Cell #53.

Sensor Number	% Above SG Median	% Below SG Median
1	26.82	72.18
2	36.43	61.15
3	22.04	73.84
4	12.31	83.10
5	38.11	54.83
6	19.55	78.10
7	4.96	93.69
8	4.91	93.77
9	36.91	61.99
10	27.08	67.82
11	53.43	35.30
12	26.17	71.61
13	43.57	44.36
14	17.40	80.51
15	9.35	89.62
16	11.18	87.57

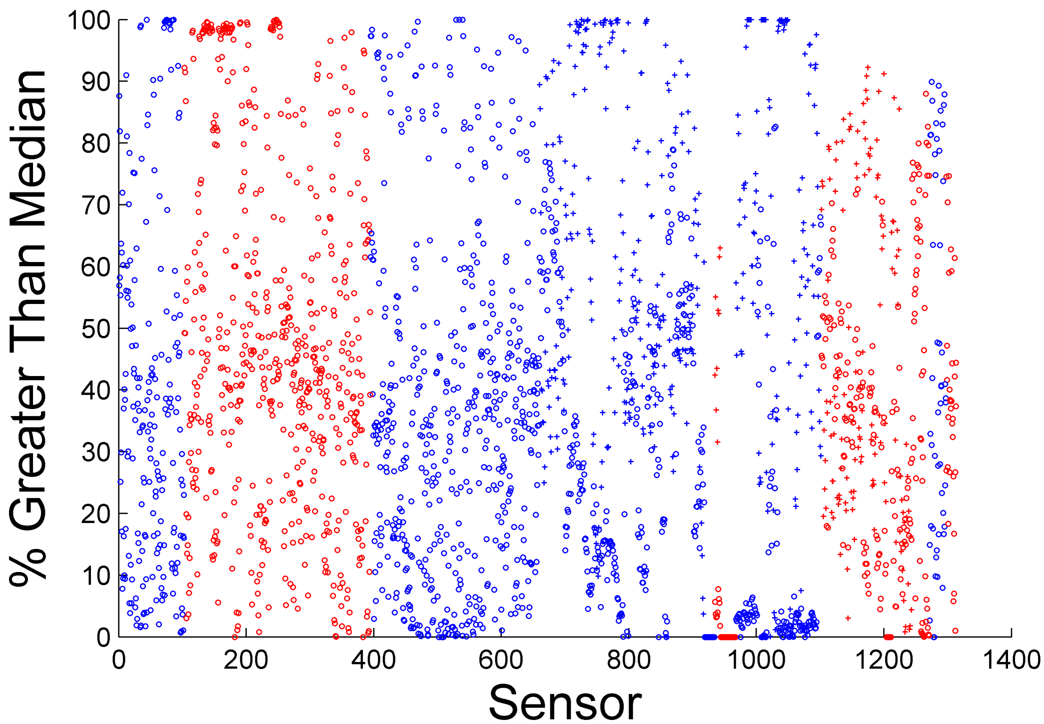

Figure 6.6: Sensor-by-sensor bias plot for the raw data: blue represent asphalt; red represent concrete; circles indicate Mainline; pluses indicate Low Volume Road.

Furthermore, it can be assumed that, as even the grouped cells are slightly different, there will be a slight cell bias that cannot be passed off as simple deviation. Also, sensors within each cell may have an individual bias. Taking this into account:

$$T(s,t) - \tilde{G}(t) = C_{bias}(s) + S_{bias}(s) + \delta \tag{6.26}$$

where

$C_{bias}(s)$ is the cell's inherant bias

$S_{bias}(s)$ is the sensor's inherant bias

$\delta(t)$ is the sensor's unbiased deviation from the median.

When comparing a sensor across its subset, if it is unbiased, the average[6] Δ would equal zero. For an unbiased sensor:

$$\frac{1}{n}\sum_{t=1}^{n}[T(s,t) - \tilde{G}(t)] = \Delta = 0 \tag{6.27}$$

For a standard, biased sensor, the average deviation without bias, δ, should still equal zero when computed across a sensor. So:

$$\frac{1}{n}\sum_{t=1}^{n}[T(s,t) - \tilde{G}(t)] = C_{bias}(s) + S_{bias}(s) \tag{6.28}$$

But the bias will be different between each sensor, so the $C_{bias}(s)+S_{bias}(s)$ can be combined into one term:

$$\frac{1}{n}\sum_{t=1}^{n}[T(s,t) - \tilde{G}(t)] = bias(s) \tag{6.29}$$

And median is preferable to mean:

$$bias(s) = \mathop{\mathbf{M}}_{t=1}^{n}[T(s,t) - \tilde{G}(t)] \tag{6.30}$$

This bias value can then be calculated for each sensor and used to counteract the effect that bias has on the subset calculations. Now, a point is flagged if:

$$\begin{cases} T(s,t) > \tilde{G}(t) + \alpha \cdot (G(t)_{[75]} - \tilde{G}(t)) + bias(t) \\ \text{or} \\ T(s,t) < \tilde{G}(t) - \alpha \cdot (\tilde{G}(t) - G(t)_{[25]}) + bias(t) \end{cases} \tag{6.31}$$

After this bias is applied to the plot of the sign test of all the sensors, the effectiveness is apparent. The data is partitioned into four sets of independent data: Mainline, asphalt (blue circles); Mainline, concrete (red circles); Low Volume Road, asphalt (blue cross); and Low Volume Road,

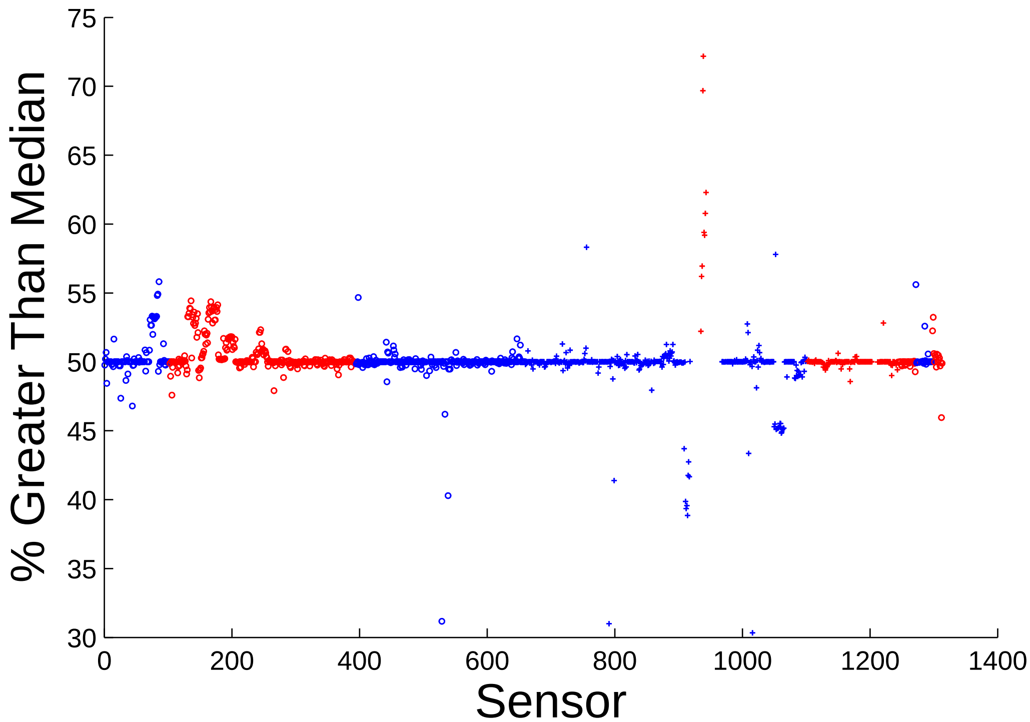

Figure 6.7: Sensor-by-sensor bias plot after applying the bias shift: blue represent asphalt; red represent concrete; circles indicate Mainline; pluses indicate Low Volume Road. This plot should be contrasted with Figure 6.6.

concrete (red cross). Figure 6.7, when compared to Figure 6.6, highlights the usefulness of the bias correction.

A good example of the usefulness of the point extremes test is Cell #4, Sensors 31 and 32. Cell #4 is on the Mainline, with an engineered section of 0.2311 m (9.1 in) of hot mix asphalt. Sensor 31 is rather shallow, at depth 0.0518 m (0.17 ft). Sensor 32 is a bit deeper, at depth 0.1250 m (0.41 ft). Both sensors are within the asphalt layer.

By themselves, the data from these two sensors appear ordinary: they seem to follow daily and annual variation without many errors (see Figures 6.8 and 6.9).

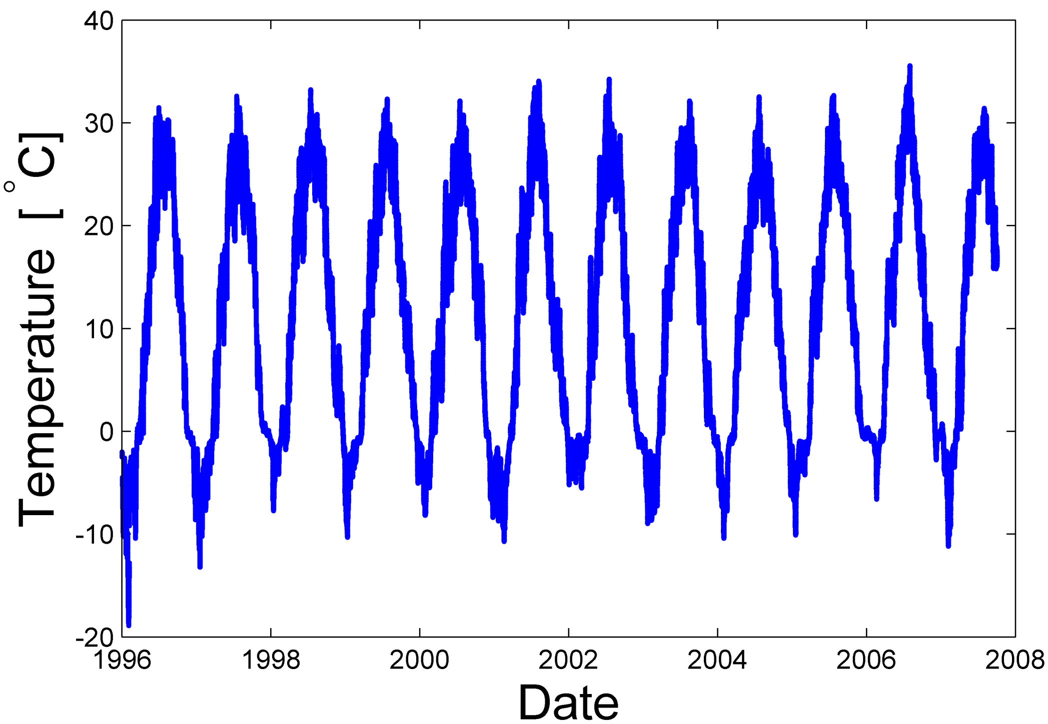

Figure 6.8: All of the considered data from Cell #4, Sensor 31. This sensor is relatively shallow, at a depth of 0.0518 m (0.17 ft) in 0.2311 m (9.1 in) of hot mix asphalt.

However, when the data from these two sensors are compared to the median and interquartile spread of their subsets[7] the sensors clearly have problems. Sensor 31 is very hypovariate while sensor 32 (a deeper sensor) is hypervariate (see Figures 6.10 and 6.11). Perhaps there was a mis-wiring[8] or data compilation error, but, whatever the reason, the point extremes test would flag it.

[6]In the process of flagging, the bias is quantified using the median. This is to "protect" the average value from erroneous extremes - and there are enough data points to make them relatively equal.

[7]Recall that these subsets include only sensors from the Mainline hot mix asphalt cells, situated at approximately the same depth.

[8]The diurnal temperature variability decreases with depth. It has been conjectured that the data stream from these two sensors have been inverted.

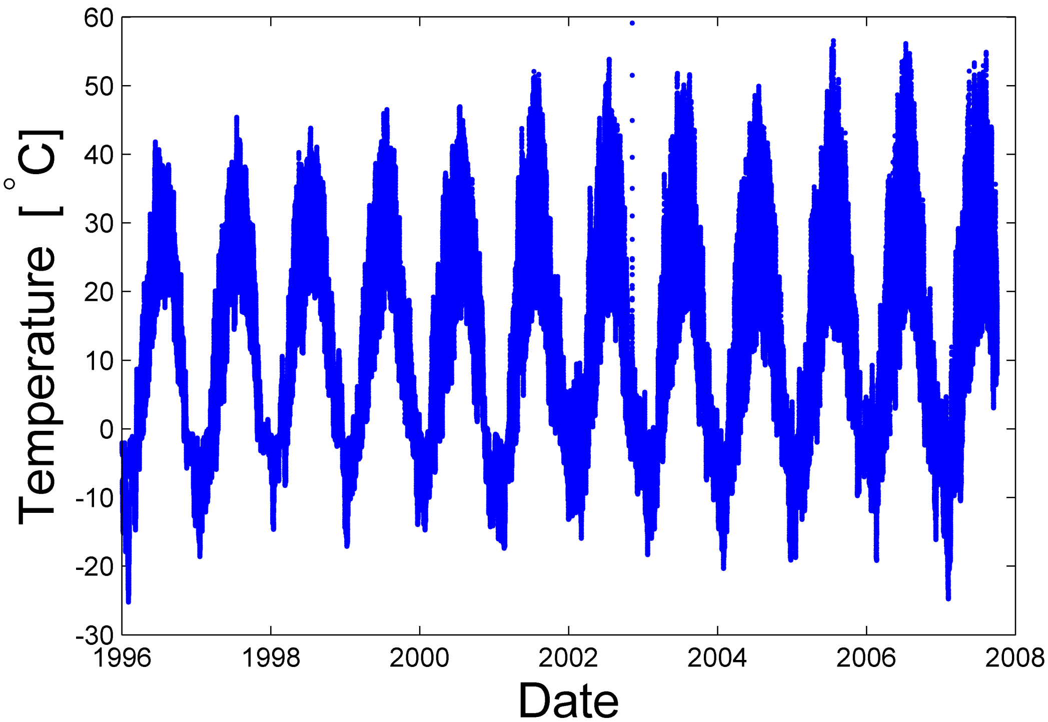

Figure 6.9: All of the considered data from Cell #4, Sensor 32. This sensor is in the middle of the engineered section, at a depth of 0.1250 $\mathtt{m}$ (0.41 $\mathtt{ft}$) in 0.2311 $\mathtt{m}$ (9.1 $\mathtt{in}$) of hot mix asphalt.

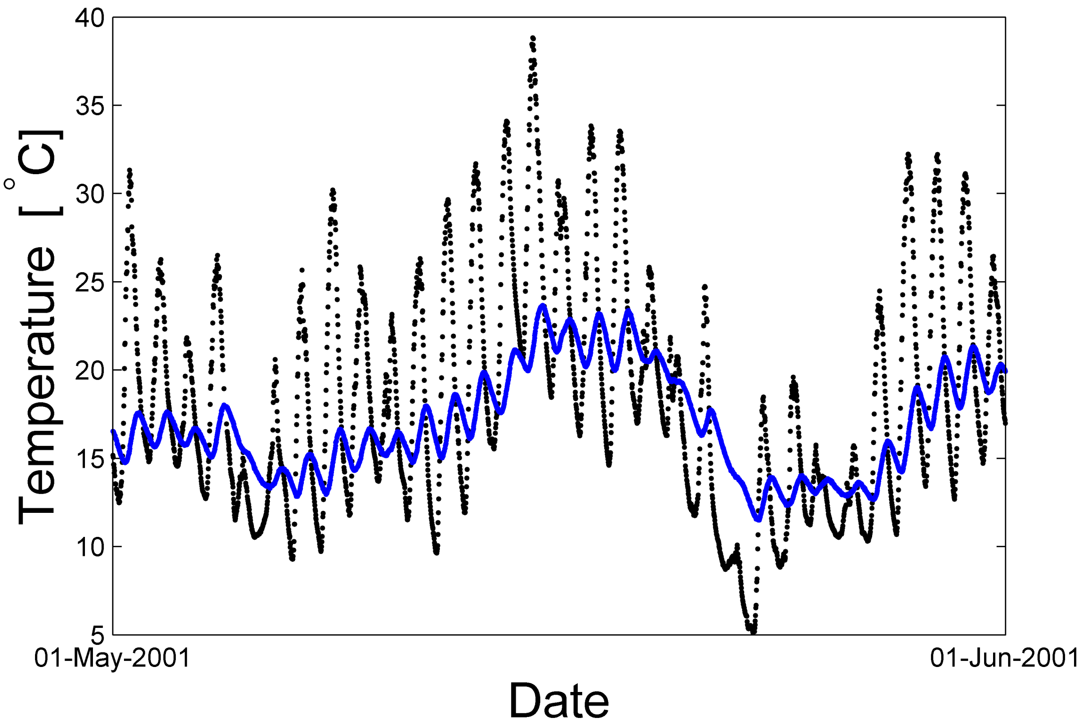

Figure 6.10: Data from May 2001. The blue line shows the recorded temperature record for Cell #4, Sensor 31. The black line shows the median temperature record for all sensors in the same subset (i.e. sensors at the same depth, on the Mainline roadway, in hot mix asphalt cells). The data from Cell #4, Sensor 31 are dramatically **less** variable (hypovariate) than would be expected; as such, they are flagged by the point extremes test.

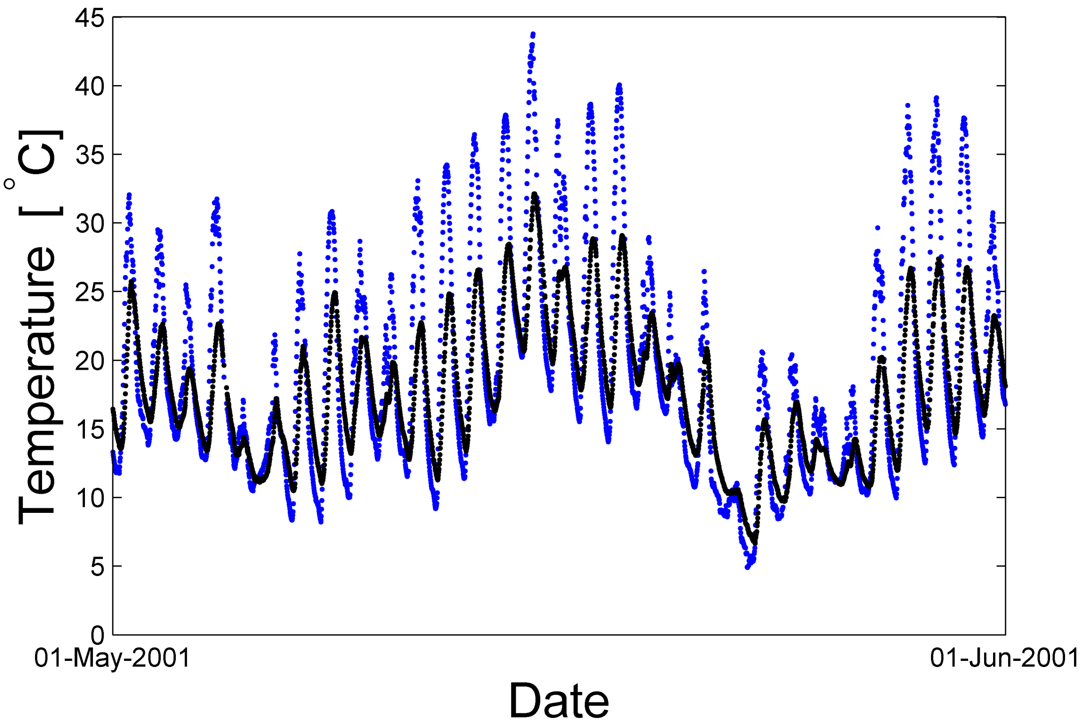

Figure 6.11: Data from May 2001. The blue line shows the recorded temperature record for Cell #4, Sensor 32. The black line shows the median temperature record for all sensors in the same subset (i.e. sensors at the same depth, on the Mainline roadway, in hot mix asphalt cells). The data from Cell #4, Sensor 32 are dramatically **more** variable (hypervariate) than would be expected; as such, they are flagged by the point extremes test.

6.11.3 `Flag_PointExtremes`

The details of the point-extremes function are show in table 6.20.

Table 6.20: The flag point extremes function.

file	`Flag_PointExtremes.m`
usage	`[] = Flag_PointExtremes()`
arguments	NONE
returns	NONE

Purpose

For each sensor, identify and flag data points deemed as "extreme" when compared to sensors in the same subset.

Preconditions

- An up-to-date "Initialization.mat" file must exist in the current working directory.
- The binary source files (i.e. `Cellxxx.mat`) must exist in the source folder identified in the initialization file.
- The following flags have already been correctly applied:
  - `FLAG_MISSING_DATA`
  - `FLAG_NOT_YET_OPERATIONAL`
  - `FLAG_DEACTIVATED`
  - `FLAG_TOO_SPARSE_DAY`
  - `FLAG_OUT_OF_RANGE`

6.12 Daily Range

6.12.1 The Flag

The details of the daily-range flag are show in table 6.21.

6.12.2 Identifying Hypervariate and Hypovariate Daily Ranges

Over the course of a day[9], there is an expected amount of temperature variation that will occur. At the most basic level, pavement will heat up during the day and cool during the night. Because of

[9] A day is defined as running from midnight to midnight.

Table 6.21: The daily range flag.

name	FLAG_DAILY_RANGE
bit number	9
binary mask	0000 0001 0000 0000
decimal value	256

this, some types of bad data can be highlighted by a lack, or excessiveness, of variation. Figure 6.12 shows a clearly hypovariate string of data points that lasts for multiple weeks. While the averages and individual temperatures are well within the range of possibility, it is erroneous nonetheless.

While checking the daily ranges against a minimum acceptable value would flag a considerable amount of data, there is a more effective way of checking the hypovariance. By comparing the daily variations to sensors in similar situations (depth, pavement type, and location[10]), a sensor can be checked for any odd variation.

The daily range flag runs through the sensor data and groups readings into days. It then finds the range over that day, as well as the range for every day in that month for every similar sensor. The sensor is flagged over the day in question if:

$$\begin{cases} R(s,d) > \tilde{R}(m) + \alpha \cdot (R_{[75]}(m) - \tilde{R}(m)) \\ \text{or} \\ R(s,d) < \tilde{R}(m) - \alpha \cdot (\tilde{R}(m) - R_{[25]}(m)) \end{cases} \tag{6.32}$$

where

- α is a predetermined multiplier[11];
- R(s,d) is the range of sensor 's' at day 'd';
- R(m) is the set of ranges of similar sensors over the month.

The month-wide range was used instead of only a day because if the comparing sensors are too similar, it may cause good data to be discarded needlessly (for example, if the upper and lower quartiles are equal to the median, any sensor that does not equal the median will be flagged). Furthermore, if there are too many bad sensors in a group, the "good data range" will be too big and may pass unwanted data. For this reason, a larger sample size is needed. A month was selected as it is long enough to buffer against bad data, but not too long to include too much seasonal variation.

6.12.3 `Flag_DailyRange`

The details of the daily-range function are show in table 6.21.

[10]The are two locations: Mainline and Low Volume Road.
[11]The default value is defined in `Initialization.mat`

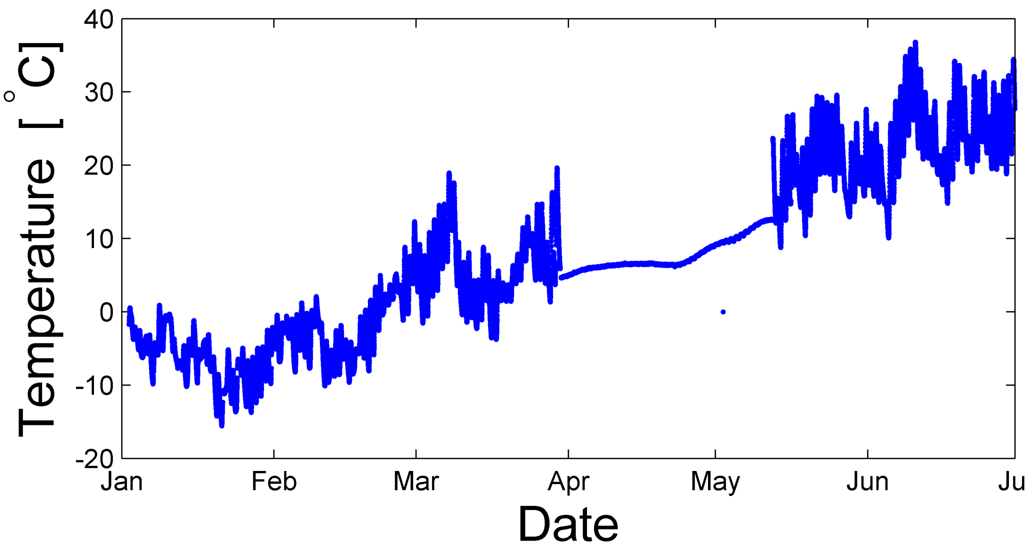

Figure 6.12: This shows the first half of 2001 for Cell #15, Sensor 3. The months of April and May present an example of hypovariate data. The overall trend through these months is correct, and the individual temperature values are reasonable (with the exception of the one aberrant point) when viewed in isolation, but the expected diurnal variation has disappeared.

Table 6.22: The flag daily range function.

file	`Flag_DailyRange.m`
usage	`[] = Flag_DailyRange()`
arguments	NONE
returns	NONE

Purpose

For each sensor, identify and flag days in which the daily range of temperatures is too large or too small when compared to similar sensors.

Preconditions

- An up-to-date "Initialization.mat" file must exist in the current working directory.
- The binary source files (i.e. `Cellxxx.mat`) must exist in the source folder identified in the initialization file.
- The following flags have already been correctly applied:
  - `FLAG_MISSING_DATA`
  - `FLAG_NOT_YET_OPERATIONAL`
  - `FLAG_DEACTIVATED`
  - `FLAG_TOO_SPARSE_DAY`
  - `FLAG_OUT_OF_RANGE`

6.13 Daily Extremes

6.13.1 The Flag

The details of the daily-extremes flag are show in table 6.23.

Table 6.23: The daily extremes flag.

name	`FLAG_DAILY_EXTREMES`
bit number	10
binary mask	0000 0010 0000 0000
decimal value	512

6.13.2 Identifying Daily Extremes

Often due to calibration errors, sensor data that appears good at first glance is actually erroneously shifted. In order to catch these faulty sensors, sensor data over a day[12] is checked against similar sensors. Similar, in this case, is defined as sensors with the same type, group, and location. After the OUT_OF_RANGE flag has been set, it can be assumed that the maximum and minimum unflagged values are within reasonable values for the sensor. So, if the daily minimum is too high or low, or the daily maximum is too high or low, all the sensor data for that day is flagged.

[12] A day is defined as running from midnight to midnight.

A daily minimum or maximum is compared to the daily minima and maxima of all similar sensors over the entire month. For example, with 30 days per month, and 30 similar sensors, we have 900 daily maxima and minima to compare.

The outlier identification, outline in Section 5.4, is applied to each day's minimum and maximum.

- The upper and lower quartile spreads are defined by

$$\text{upper quartile spread} = \text{75th percentile} - \text{50th percentile} \tag{6.33}$$
$$\text{lower quartile spread} = \text{50th percentile} - \text{25th percentile} \tag{6.34}$$

- The upper and lower bounds for the month's minima and maxima are defined[13] as

$$\text{upper bound} = \text{median} + \zeta \cdot \text{upper quartile spread} \tag{6.35}$$
$$\text{lower bound} = \text{median} - \zeta \cdot \text{lower quartile spread} \tag{6.36}$$

  where ζ is defined such that

$$\boldsymbol{Pr}(\mathtt{r(t)} < \text{lower bound}) = \mathtt{PROBABILITY\_OF\_FALSE\_POSITIVE/2} \tag{6.37}$$
$$\boldsymbol{Pr}(\mathtt{r(t)} > \text{upper bound}) = \mathtt{PROBABILITY\_OF\_FALSE\_POSITIVE/2} \tag{6.38}$$

  where `PROBABILITY_OF_FALSE_POSITIVE` is a manifest constant defined in the initialization[14] function. The current value for this constant is:

$$\mathtt{PROBABILITY\_OF\_FALSE\_POSITIVE} = 1/100000 \tag{6.39}$$

- Any daily minimum or maximum exceeding the upper or lower bounds are identified as outliers and the associated day is flagged as extreme.

6.13.3 `Flag_DailyExtremes`

The details of the daily-extremes flag are show in table 6.24.

Table 6.24: The flag daily extremes function.

file	`Flag_DailyExtremes.m`
usage	`[] = Flag_DailyExtremes()`
arguments	NONE
returns	NONE

[13]This definition uses a Normal distribution model, which was selected based upon a study of the maxima from a representative subset of the data.

[14]See `CreateInitialization.m`

Purpose

For each sensor, identify and flag days in which the maximum and minimum values are "extreme" when compared to similar sensors.

Preconditions

- An up-to-date "Initialization.mat" file must exist in the current working directory.
- The binary source files (i.e. `Cellxxx.mat`) must exist in the source folder identified in the initialization file.
- The following flags have already been correctly applied:
  - `FLAG_MISSING_DATA`
  - `FLAG_NOT_YET_OPERATIONAL`
  - `FLAG_DEACTIVATED`
  - `FLAG_TOO_SPARSE_DAY`
  - `FLAG_OUT_OF_RANGE`

6.14 Intermittent Data

6.14.1 The Flag

The details of the intermittent-data flag are show in table 6.25.

Table 6.25: The intermittent data flag.

name	`FLAG_INTERMITTENT_DATA`
bit number	10
binary mask	0000 0100 0000 0000
decimal value	1024

6.14.2 Identifying Intermittent Data

In order to ensure the validity of the data, the "'better safe then sorry"' approach is often the best. While the flagging functions are reliable, it may happen, occasionally that a bad reading remains unflagged. This is usually because the data, while bad, is well within the realm of possibility. It is only until the bad datum is closely inspected visually that its error is apparent. Because of this, it is important to find and flag problematic sections of data. The `Flag_IntermittentData` flag works much like the Missing Data Intermittency flag - If there are any flagged data points (with any of the 16 flags) within 6 hours of each other[15] then all of the data in between the two are flagged.

[15] At the default setting as defined by MINIMUM_INTERMITTENT_TIME in `Initialization.mat`

6.14.3 Flag_IntermittentData

The details of the intermittent-data function are show in table 6.26.

Table 6.26: The flag intermittent data function.

file	Flag_IntermittentData.m
usage	[] = Flag_IntermittentData()
arguments	NONE
returns	NONE

Purpose

For each sensor, identify and flag data that is in between two close, bad data points.

Preconditions

- An up-to-date "Initialization.mat" file must exist in the current working directory.
- The binary source files (i.e. Cellxxx.mat) must exist in the source folder identified in the initialization file.
- The following flags have already been correctly applied:
  - FLAG_MISSING_DATA
  - FLAG_NOT_YET_OPERATIONAL
  - FLAG_DEACTIVATED
  - FLAG_TOO_SPARSE_DAY
  - FLAG_OUT_OF_RANGE
  - FLAG_NEIGHBORHOOD_OUTLIERS
  - FLAG_LAG_ONE_OUTLIERS
  - FLAG_POINT_EXTREMES
  - FLAG_DAILY_RANGE
  - FLAG_DAILY_EXTREMES

6.15 Inconsistent Day

6.15.1 The Flag

The details of the inconsistent-day flag are show in table 6.27.

Table 6.27: The inconsistent day flag.

name	FLAG_INCONSISTENT_DAY
bit number	12
binary mask	0000 1000 0000 0000
decimal value	2048

6.15.2 Identifying Inconsistent Days

Even with completely nonsensical data from a clearly dysfunctional sensor, there is always a chance that a reading may "slip through the cracks" of the flagging dragnet. If a seemingly admissible point is surrounded by incongruous data, it is reasonable to assume that it is bad as well. The "Inconsistent Day" flag breaks sensor data into 24 hour groups (about 96 readings if consistent) - from midnight to midnight. For each of these days, if there are too few unflagged data points[16] or if the ratio of unflagged points to total points is too small[17] the entire day is flagged.

While this process may occasionally flag correct data points, it should have very few adverse effects on the final data set. As these good points are surrounded by bad data, they lack a frame of reference needed for them to be useful in analysis. Furthermore, having full sections of data missing is much easier to conceptualize than a few points sprinkled about an otherwise empty period.

The analysis for this flag is carried out on a sensor-by-sensor basis. When identifying inconsistent days for each sensor, only data from that specific sensor's time series are used.

6.15.3 `Flag_InconsistentDay`

The details of the inconsistent-day function are show in table 6.28.

Table 6.28: The flag inconsistent day function.

file	Flag_InconsistentDay.m
usage	[] = Flag_InconsistentDay()
arguments	NONE
returns	NONE

Purpose

For each sensor, identify and flag days with too few unflagged sensor readings.

[16]Default 10 data points as defined by MINIMUM_GOOD_DATA_DAY in "Initialization.mat"

[17]Default ratio of 0.5 as defined by MINIMUM_FRACTION_GOOD_DATA_DAY in "Initialization.mat"

Preconditions

- An up-to-date ”Initialization.mat” file must exist in the current working directory.
- The binary source files (i.e. `Cellxxx.mat`) must exist in the source folder identified in the initialization file.
- The following flags have already been correctly applied:
  - FLAG_MISSING_DATA
  - FLAG_NOT_YET_OPERATIONAL
  - FLAG_DEACTIVATED
  - FLAG_TOO_SPARSE_DAY
  - FLAG_OUT_OF_RANGE
  - FLAG_NEIGHBORHOOD_OUTLIERS
  - FLAG_LAG_ONE_OUTLIERS
  - FLAG_POINT_EXTREMES
  - FLAG_DAILY_RANGE
  - FLAG_DAILY_EXTREMES
  - FLAG_INTERMITTENT_DATA

6.16 Inconsistent Week

6.16.1 The Flag

The details of the inconsistent-week flag are show in table 6.29.

Table 6.29: The inconsistent week flag.

name	FLAG_INCONSISTENT_WEEK
bit number	13
binary mask	0001 0000 0000 0000
decimal value	4096

6.16.2 Identifying Inconsistent Weeks

Similar to the inconsistent-day flag, the “Inconsistent Week” flag breaks sensor data into 7 day groups (about 672 readings if consistent) - from Sunday to Sunday. For each of these weeks, if there are too few unflagged data points[18] or if the ratio of unflagged points to total points is too small[19] the entire week is flagged.

While this process may occasionally flag correct data points, it should have very few adverse effects on the final data set. As these good points are surrounded by bad data, they lack a frame of reference needed for them to be useful in analysis. Furthermore, having full sections of data missing is much easier to conceptualize than a few points sprinkled about an otherwise empty period.

The analysis for this flag is carried out on a sensor-by-sensor basis. When identifying inconsistent weeks for each sensor, only data from that specific sensor's time series are used.

6.16.3 `Flag_InconsistentWeek`

The details of the inconsistent-week function are show in table 6.30.

Table 6.30: The flag inconsistent week function.

file	`Flag_InconsistentWeek.m`
usage	`[] = Flag_InconsistentWeek()`
arguments	NONE
returns	NONE

Purpose

For each sensor, identify and flag weeks with too few unflagged sensor readings.

Preconditions

- An up-to-date "Initialization.mat" file must exist in the current working directory.
- The binary source files (i.e. `Cellxxx.mat`) must exist in the source folder identified in the initialization file.
- The following flags have already been correctly applied:
  - `FLAG_MISSING_DATA`
  - `FLAG_NOT_YET_OPERATIONAL`
  - `FLAG_DEACTIVATED`
  - `FLAG_TOO_SPARSE_DAY`
  - `FLAG_OUT_OF_RANGE`
  - `FLAG_NEIGHBORHOOD_OUTLIERS`

[18]Default 70 data points as defined by MINIMUM_GOOD_DATA_WEEK in "Initialization.mat"
[19]Default ratio of 0.5 as defined by MINIMUM_FRACTION_GOOD_DATA_WEEK in "Initialization.mat"

- FLAG_LAG_ONE_OUTLIERS
- FLAG_POINT_EXTREMES
- FLAG_DAILY_RANGE
- FLAG_DAILY_EXTREMES
- FLAG_INTERMITTENT_DATA
- FLAG_INCONSISTENT_DAY

6.17 Inconsistent Month

6.17.1 The Flag

The details of the inconsistent-month flag are show in table 6.31.

Table 6.31: The inconsistent month flag.

name	FLAG_INCONSISTENT_MONTH
bit number	14
binary mask	0010 0000 0000 0000
decimal value	8192

6.17.2 Identifying Inconsistent Months

Similar to the inconsistent-day and inconsistent-week flags, the "Inconsistent Month" flag breaks sensor data into groups by month (about 2900 readings if consistent). For each of these months, if there are too few unflagged data points[20] or if the ratio of unflagged points to total points is too small[21] the entire month is flagged.

While this process may occasionally flag correct data points, it should have very few adverse effects on the final data set. As these good points are surrounded by bad data, they lack a frame of reference needed for them to be useful in analysis. Furthermore, having full sections of data missing is much easier to conceptualize than a few points sprinkled about an otherwise empty period.

The analysis for this flag is carried out on a sensor-by-sensor basis. When identifying inconsistent months for each sensor, only data from that specific sensor's time series are used.

6.17.3 `Flag_InconsistentMonth`

The details of the inconsistent-month function are show in table 6.32.

[20] Default 280 data points as defined by MINIMUM_GOOD_DATA_MONTH in "Initialization.mat"

[21] Default ratio of 0.5 as defined by MINIMUM_FRACTION_GOOD_DATA_MONTH in "Initialization.mat"

Table 6.32: The flag inconsistent month function.

file	`Flag_InconsistentMonth.m`
usage	`[] = Flag_InconsistentMonth()`
arguments	NONE
returns	NONE

Purpose

For each sensor, identify and flag months with too few unflagged sensor readings.

Preconditions

- An up-to-date ”Initialization.mat” file must exist in the current working directory.
- The binary source files (i.e. `Cellxxx.mat`) must exist in the source folder identified in the initialization file.
- The following flags have already been correctly applied:
  - `FLAG_MISSING_DATA`
  - `FLAG_NOT_YET_OPERATIONAL`
  - `FLAG_DEACTIVATED`
  - `FLAG_TOO_SPARSE_DAY`
  - `FLAG_OUT_OF_RANGE`
  - `FLAG_NEIGHBORHOOD_OUTLIERS`
  - `FLAG_LAG_ONE_OUTLIERS`
  - `FLAG_POINT_EXTREMES`
  - `FLAG_DAILY_RANGE`
  - `FLAG_DAILY_EXTREMES`
  - `FLAG_INTERMITTENT_DATA`
  - `FLAG_INCONSISTENT_DAY`
  - `FLAG_INCONSISTENT_MONTH`

Chapter 7

Lag One Analysis

7.1 Lag One: Outline

A good way to check the reliability of data, as well as find information about the temporal gradients, is to run it through a difference operator. A difference operator, or in this case, a backward difference operator, is a function that finds the difference between the value of a data point and the value of the point directly preceding it. More specifically:

$$\Delta f(t) = f(t) - f(t-1) \tag{7.1}$$

This process of comparing a datum point to the point directly before it is known as a lag-one test. While there are difference operators that look farther back in time than one time-step in order to dampen small variations, lag-one is the simplest and most fitting for use in MnROAD analysis.

7.2 Lag One Implementation

The process in which the lag-one statistics for a sensor are calculated is a fairly simple one. The function marches through the sensor data, from the earliest reading to the latest, calculating the difference between each temperature reading and the one before it. For example:

Table 7.1: An example lag-one calculation.

Time	Reading	Lag One
-	°C	°C
1:15	10.0	-
1:30	11.5	1.5
1:45	12.5	1.0
2:00	11.5	-1.0
2:15	11.5	0.0

As is apparent, the lag-one value for the first reading in a sensor cannot be calculated as it has no predecessor. This is insignificant as there are enough readings in every sensor to supply sufficient statistics for analysis. However, this does become an issue for flagged or missing data. If either the reading in question or the one before it are bad, the lag-one value will either be bad or missing. While there are many different ways to deal with this problem, the simplest is to simply ignore any flagged data or any data following a flagged data point. For example:

Table 7.2: An example lag-one calculation with missing data.

Time	Reading	Lag One
-	°C	°C
1:15	10.0	-
1:30	11.5	1.5
1:45	Missing	-
2:00	11.5	-
2:15	11.5	0

In the full scale calculations, the sheer number of good readings outweighs the ignored lag-one values.

7.3 Lag One Distribution

In running the lag-one function on all of the sensors, a series of lag-one arrays are produced. These can be used to highlight erroneous data as well as learn valuable information about the temporal gradients of different sensors.

The first step in analyzing this new-found data is to look at its histogram and check if it follows a standard distribution. At first glance, Figure 7.1 appears to be following a normal distribution. If the lag-one distribution is normal, it allows for a much simpler analysis process. In order to test the normality of the lag-one data, it is standard practice to construct a normal probability plot.

In a normal probability plot, the ordered statistics (in this case, the lag-one values in order from lowest to highest) are normalized[1] and plotted against a standard normal. If the distribution is normal, the normal probability plot should form a reasonably straight line. The less straight the line, the less normal the distribution.

Figure 7.2 disproves the normality of the lag-one statistics. The tails of the distribution are far too long. Furthermore, the kurtosis (peakedness) of the data is 6.49, where normally distributed data tends to have a kurtosis of around three.

From experience, the lag-one data looks like it would fit a generalized normal distribution. The generalized normal distribution follows the probability density function of:

$$\frac{\beta}{2\alpha\Gamma(1/\beta)}e^{-(|x-\mu|/\alpha)^{\beta}} \tag{7.2}$$

[1]The mean is subtracted from each value, and that difference is divided by the standard deviation. This will center the distribution around zero and scale the values to the same magnitude as the standard normal.

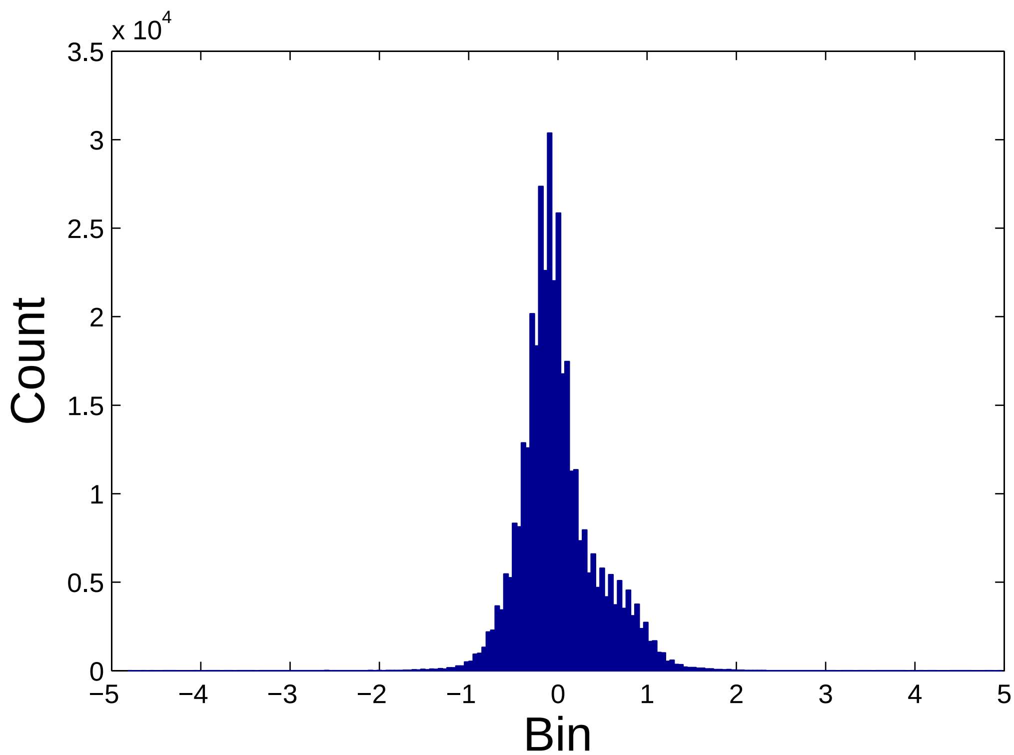

Figure 7.1: A representative histogram of lag-one data from Cell #5, Sensor 1. The histogram is roughly "bell-shaped", but there is a clear asymmetry.

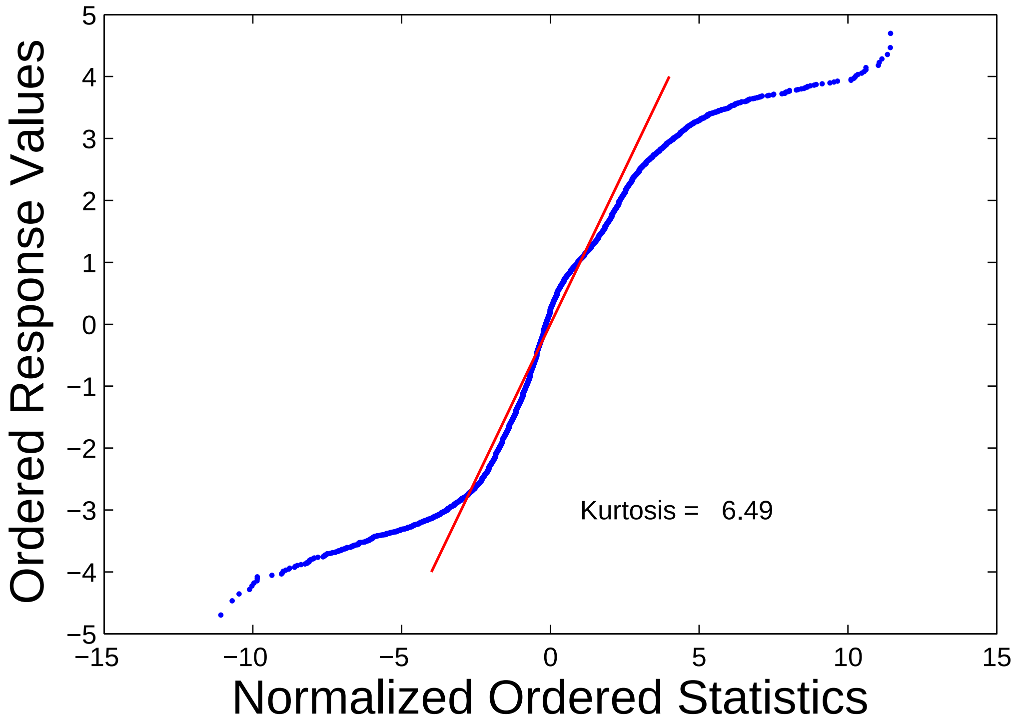

Figure 7.2: A representative Normal probability plot of lag-one data for Cell #5, Sensor 1. The red line represents a theoretical Normal distribution, the blue dots are the measured data.

where

- Γ represents the gamma function,
- μ denotes the location,
- α denotes the scale, and
- β denotes the shape.

Using a optimization function, μ, α, and β can be calculated to make a best-fit distribution. For Cell #5, Sensor 1, these values were decided to be $\mu = -0.066$, $\alpha = 0.318$, and $\beta = 1.014$. The visual representation of this fit can be seen in Figure 7.3.

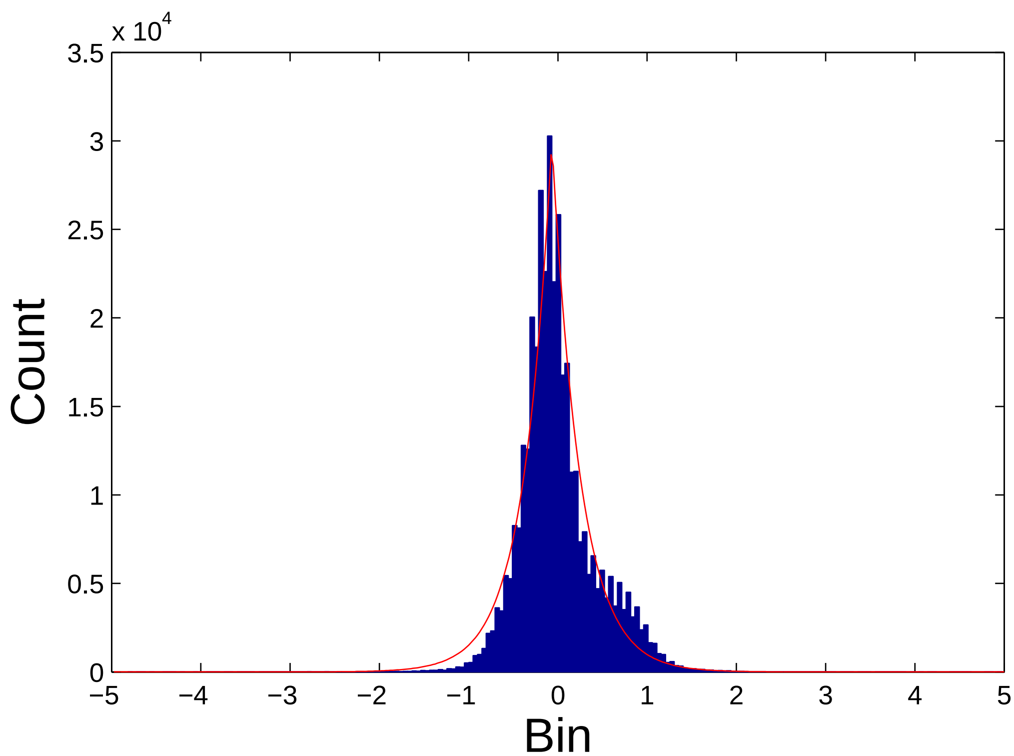

Figure 7.3: A representative Generalized Normal distribution fit for lag-one data from Cell #5, Sensor 1. The red line represents a theoretical Normal distribution, the blue bars are the measured data.

The distribution in Figure 7.3 seems to be a good one. This will aid in assumptions and decisions in the future.

7.4 Lag One Versus Time

An important facet of lag-one statistics is how they vary over time. This can provide valuable information about temporal gradients with respect to time of day, season, and year. Figure 7.4

shows that the temporal variation of lag-one values is much higher during the summer months than the winter. The red lines represent the 1st and 99th percentiles of the data. After a quick inspection of this plot, there is clearly erroneous data during the winter of 1999[2].

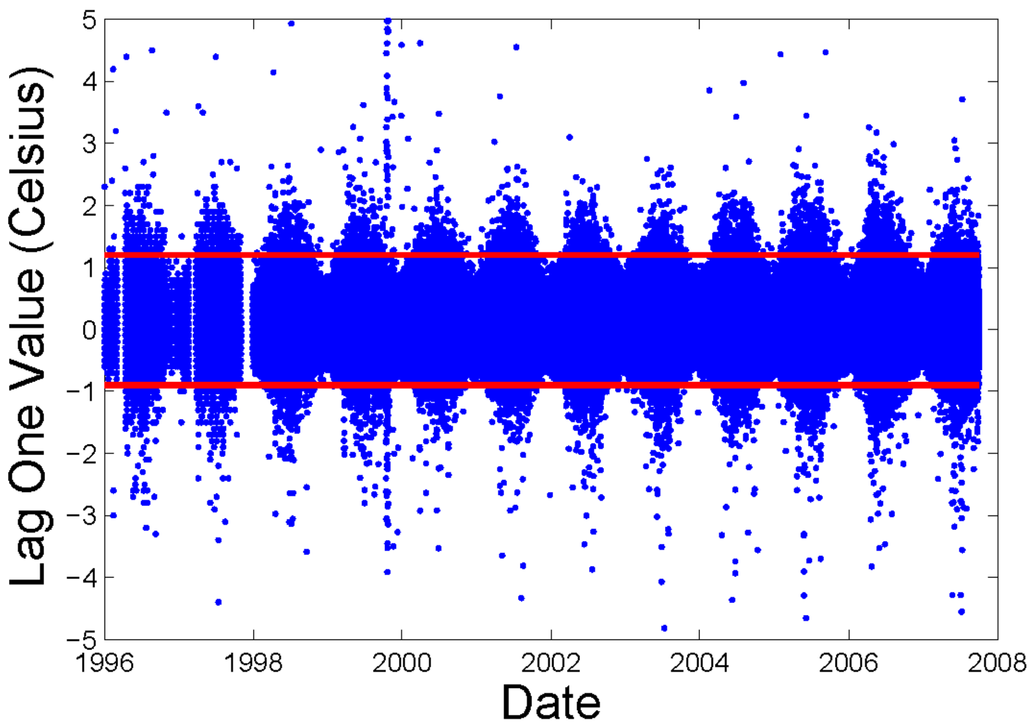

Figure 7.4: A representative plot the Lag-one-versus-time data from Cell #5, Sensor 1. The blue dots show the measured data; the red lines depict the 1st and 99th percentiles of the data.

7.5 Lag One Versus Temperature

Along with time, finding how the temperature variance is affected by the temperature itself is extremely important. By plotting lag-one values vs. temperature for a sensor, these effects are clear. Figure 7.5 shows that, while fairly uniform, the temperature variance is positively correlated with the temperature.

Conversely, the deeper the sensor, the more uniform the lag-one vs. temperature plot becomes. In Figure 7.6 the lag-one values and temperature are plotted for a sensor that is 4.93 feet below the surface.

[2]Often, the easiest way to identify bad data is to find areas of data that look "unnatural" when plotted. This area, for example, has too many extreme values, both high and low, in too short of time.

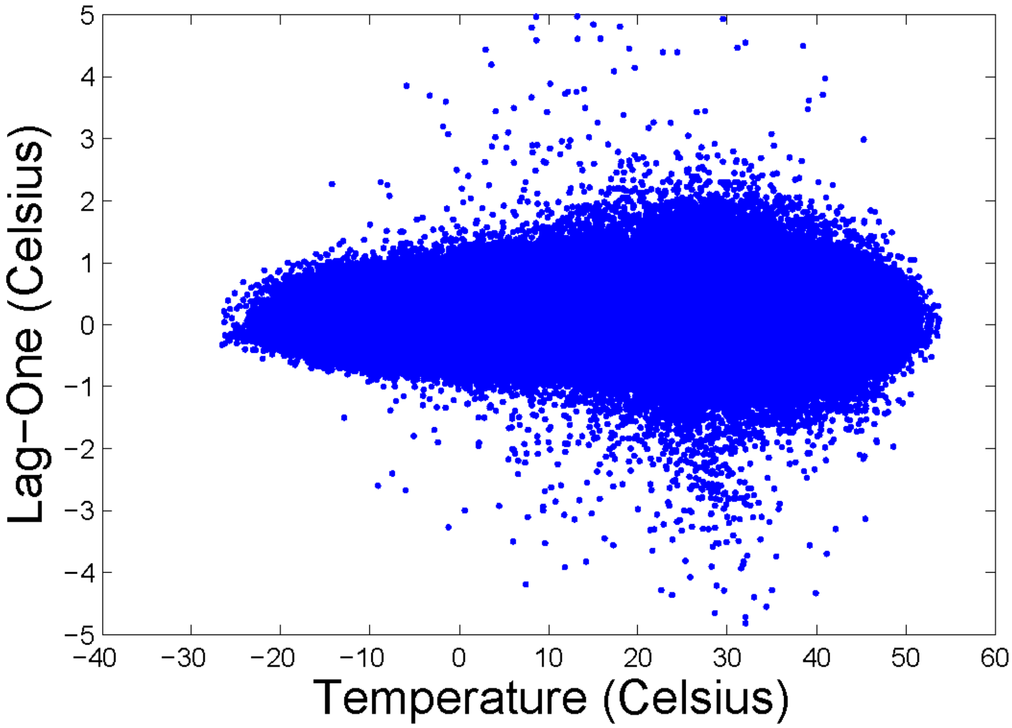

Figure 7.5: Lag one versus temperature for Cell #5, Sensor 1.

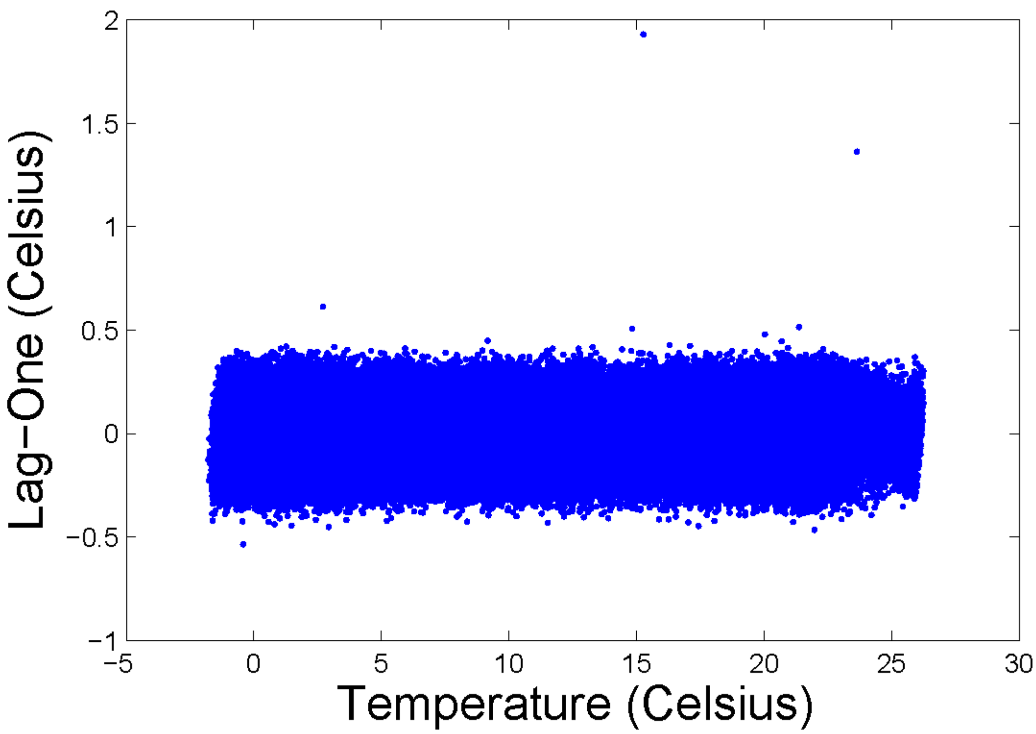

Figure 7.6: Lag one versus temperature for Cell #5, Sensor 11.

Chapter 8

Summary Analysis

8.1 Some Examples

8.1.1 Cell #1, Sensor 4

Figure 8.1 presents[1] all flags for Cell #1, Sensor 4. The plot title includes:

- Cell number
- Sensor number
- Subset(type, group, loc, tree)
  - type: HMA, PCC, Other
  - group: 1,2, ..., 13
  - loc: ML = Mainline, LV = Low Volume Road
  - tree: sensor trees are numbered sequentially within the cell
- Unflagged(`PercentActiveGood`, `PercentTotalGood`)
  - `PercentActiveGood` percentage of the data from active sensors that are unflagged.
  - `PercentTotalGood` percentage of the total data that are unflagged

The abscissa shows the calendar date, while the ordinate segregates the flag bits. Each red circle indicates a set flag bit at the associated date. The blue dots at flag number 0 (plotted along the abscissa) indicate dates at which no flags bits were set.

As indicated by the visible thick red bars, and the absence of any blue dots along the abscissa, this sensor was inactive before January 1998. There are five red bars shown:

- Flag 1: `FLAG_MISSING_DATA` identifies missing data, which is designated as such in the original raw data files.

[1]The plot was generated by the helper routine `PlotFlags.m`.

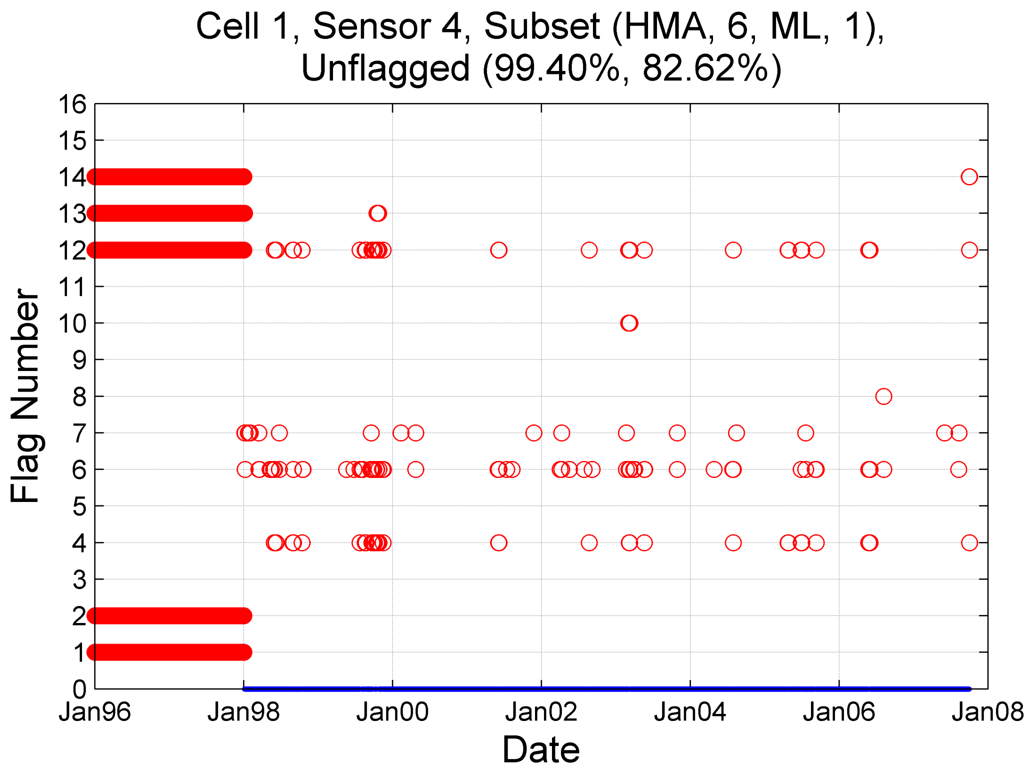

Figure 8.1: A graphical presentation of all flags for Cell #1, Sensor 4.

- Flag 2: `FLAG_NOT_YET_OPERATIONAL` identifies that the missing data is because the sensor has yet to be turned on.
- Flag 12: `FLAG_INCONSISTENT_DAY` indicates that these days lack sufficient unflagged data for any of the data to be considered reliable. This flag adds no additional information in this case.
- Flag 13: `FLAG_INCONSISTENT_WEEK` indicates that these weeks lack sufficient unflagged data for any of the data to be considered reliable. This flag adds no additional information in this case.
- Flag 14: `FLAG_INCONSISTENT_MONTH` indicates that these months lack sufficient unflagged data for any of the data to be considered reliable. This flag adds no additional information in this case.

After January 1998 this sensor is operational, and functioning remarkably well. As reported in the plot title, 99.40% of the data records are unflagged for this sensor, once the sensor becomes operational. The consistent high quality data from this sensor is further depicted by the apparent blue bar along the abscissa. A blue dot is plotted at every unflagged day. The sequence of dots appears to be a line because there are so few gaps and the gaps that exist are too small to see.

Nonetheless, some flags are set after the sensor becomes operational after January 1998:

- Flag 4: `FLAG_TOO_SPARSE_DAY` indicates that these days lack sufficient active data for any of the data to be considered reliable.

- Flag 6: `FLAG_NEIGHBORHOOD_OUTLIERS` indicates data that are significantly different from the associated local moving average.
- Flag 7: `FLAG_LAG_ONE_OUTLIERS` indicates data that have too large a jump from one measurement to he next. Often, but not always, flags 6 and 7 apply to the same records.
- Flag 10: `FLAG_DAILY_EXTREMES` indicates that the range of measurements over this day are significantly different from the range of measurements over the same day from sensors in the same (type,depth,location) subset. In this case, there are only a few days in the Winter of 2003 that get flagged.
- Flag 12: `FLAG_INCONSISTENT_DAY` indicates that these days lack sufficient unflagged data for any of the data to be considered reliable. This is different from flag 4 in that this flag considers all other flags and not just missing data.
- Flag 13: `FLAG_INCONSISTENT_WEEK` indicates that these weeks lack sufficient unflagged data for any of the data to be considered reliable.
- Flag 14: `FLAG_INCONSISTENT_MONTH` indicates that these months lack sufficient unflagged data for any of the data to be considered reliable.

Figure 8.2 presents[2] all temperature records, with colored flags, for Cell #1, Sensor 4.

A cursory inspection of Figure 8.2 may lead to the conclusion that this is a faulty, generally unreliable, sensor. This is not the case. More than 99.4% of the data records, after the sensor becomes operational in January 1998, are wholly unflagged. Nonetheless, there are some erroneous, questionable, or exceptional data in this sensor's data stream. An enlargement of late September 1999 is shown in Figure fig:C1S4tempssubset.

8.1.2 "Dead Data" – Cell #15, Sensor 3

Figure 3.5 shows a data from Cell #15, Sensor 3, for a span of about 40 days, during which the anticipated diurnal and meteorological variations disappear. These "dead data" occur in Spring 2000. Figure 8.4 presents all flags for Cell #1, Sensor 4. The red blotch for Flags 8 and 9 in Spring 2000 are a consequence of these 40 days of dead data.

8.1.3 "Defective Sensor" – Cell #93, Sensor 1

Figure 3.8 shows part of the data stream for Cell #93, Sensor 1. The figure leads to the conclusion that the sensor is defective and should be retired. Figure 8.5 presents all flags for Cell #93, Sensor 1. This figure supports the previous conclusion.

8.2 Data Quality Summary Statistics

The considered data set spanned 1 January 1996 through 31 December 2007. The complete summary counts are given in Table 8.1. The "Total number of data records" and the "Total number of

[2] The plot, and all comparable plots in this section, were generated by the helper routine `PlotTemperatures.m`.

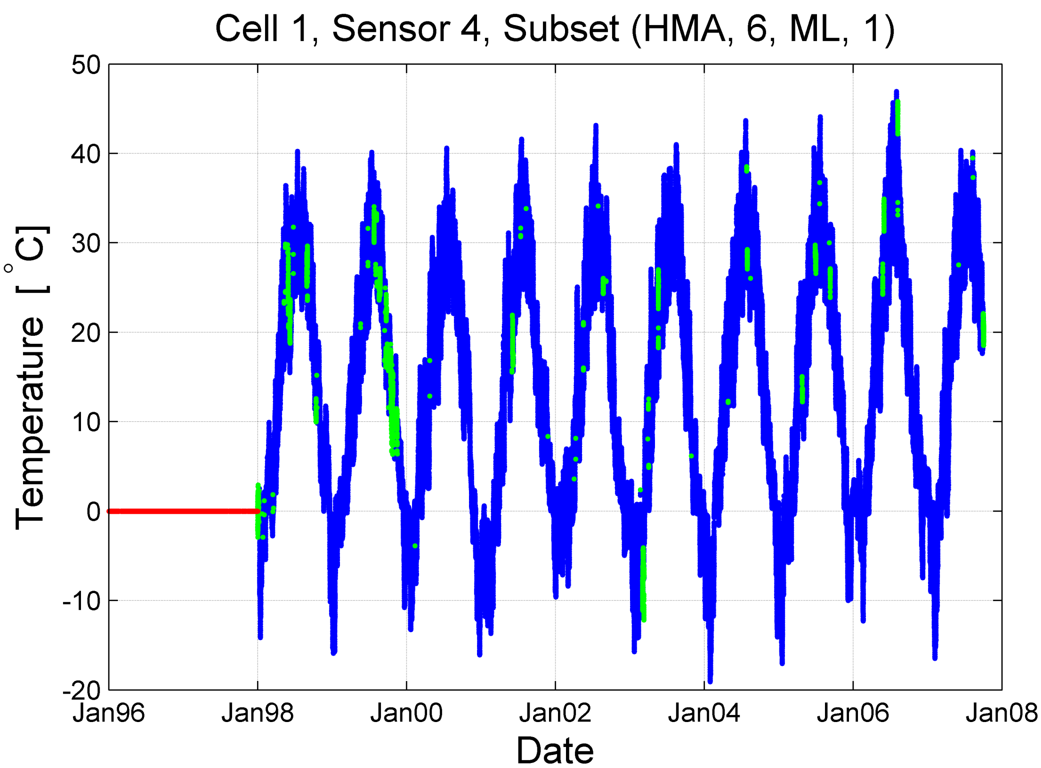

Figure 8.2: A graphical presentation of all temperature records, with colored flags, for Cell #1, Sensor 4. Red dots indicate missing data, green dots show flagged data, and blue dots are unflagged data.

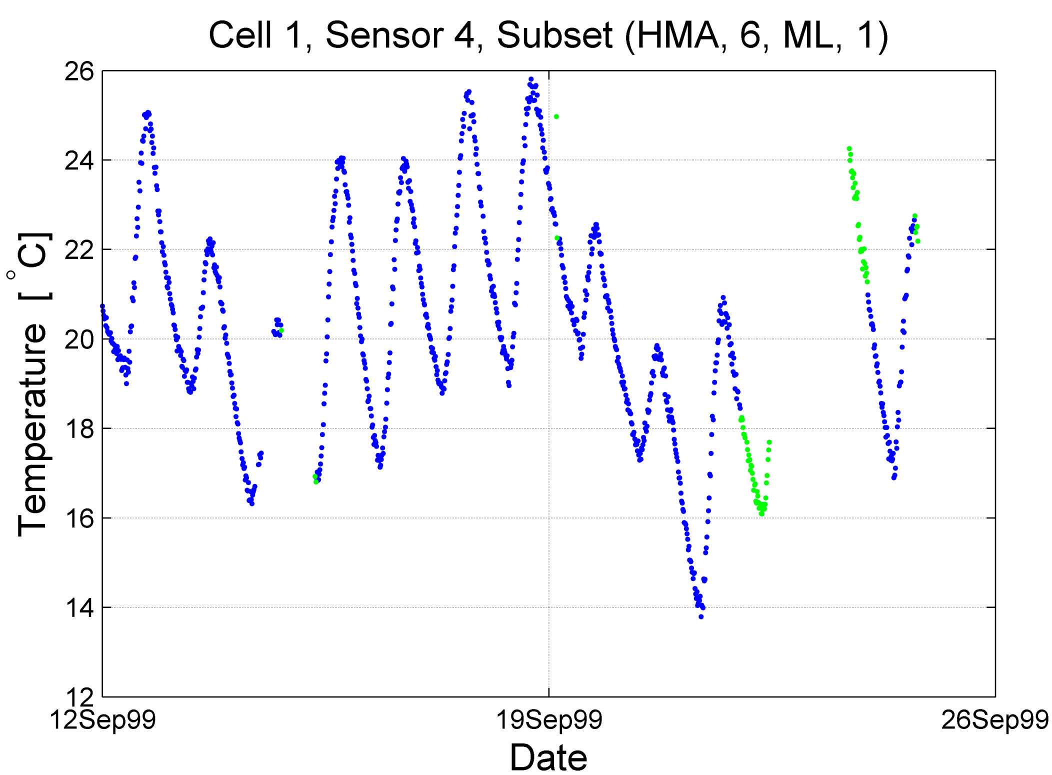

Figure 8.3: A graphical presentation of temperature records from late September 1999, with colored flags, for Cell #1, Sensor 4. Green dots show flagged data, and blue dots are unflagged data.

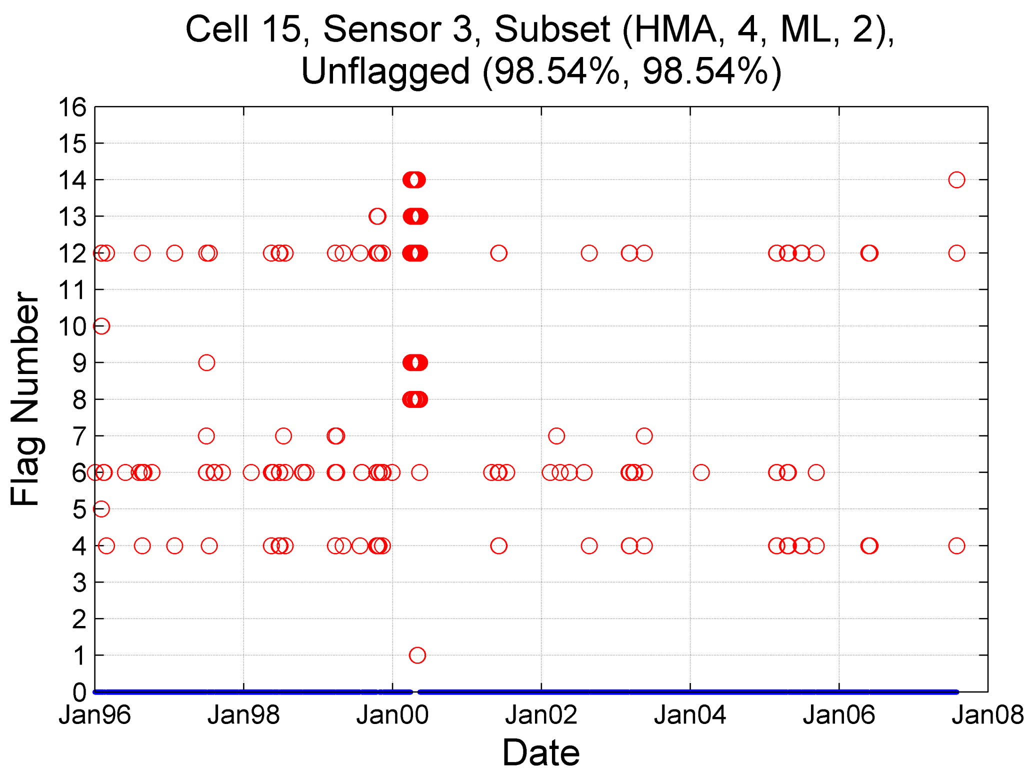

Figure 8.4: A graphical presentation of all flags for Cell #15, Sensor 3. Flags 8 and 9 are set in Spring 2000 as a consequence of the "dead data" shown in Figure 3.5.

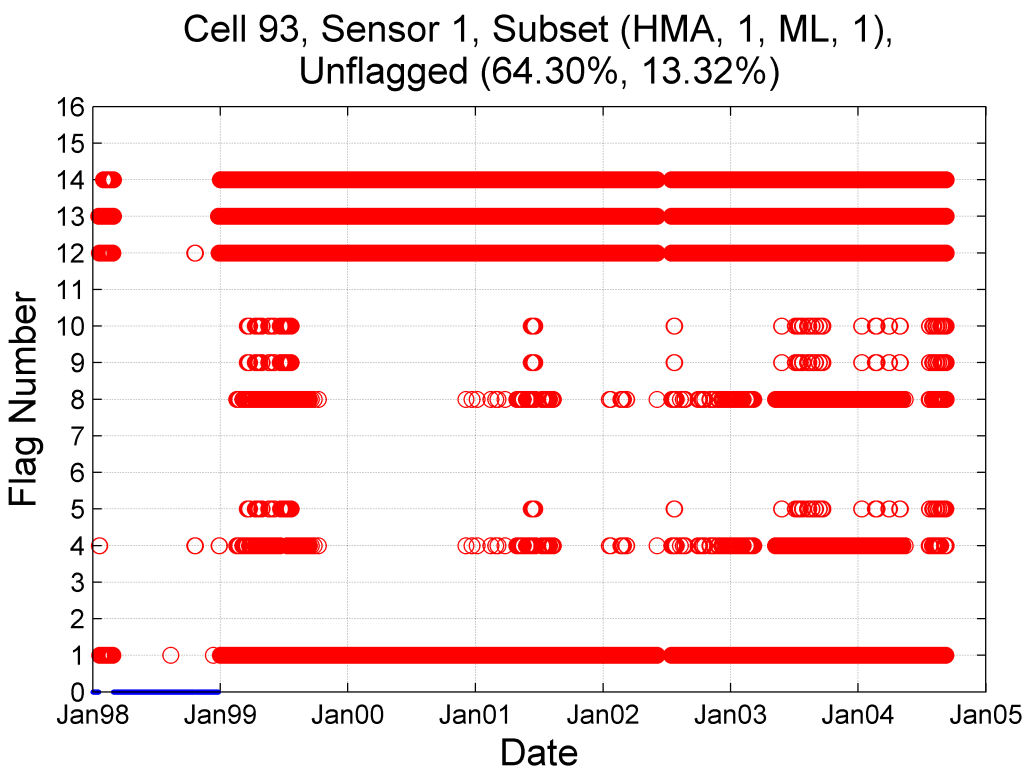

Figure 8.5: A graphical presentation of all flags for Cell #93, Sensor 1. This figure reinforces the conclusion drawn from Figure 3.8: the sensor is defective and should be retired.

flagged data records" include all of the missing data before sensors became operational and after sensor were taken out of service. The more useful numbers are the "Total number of active data records" and the "Total number of unflagged data records". The active data records include only data from those sensors that are nominally on-line and operational. The unflagged data records are data there passed all of the tests: i.e. they are "good" data.

Table 8.1: Summary count statistics for the entire considered data set.

Total number of cells	48
Total number of sensors	$1,313$
Total number of data records	$471,178,324$
Total number of active data records	$340,361,568$
Total number of flagged data records	$146,748,725$
Total number of unflagged data records	$324,429,599$

Thus, overall the percent of the data from nominally on-line and operational sensors that are "good" is

$$\frac{324,429,599}{340,361,568} = 0.9532 \tag{8.1}$$

or about 95.3%.

The "percent good" may be computed in a similar fashion for each of the $1,313$ sensors:

$$\text{percent good} = 100 \times \frac{\text{number of unflagged data records}}{\text{number of active data records}} \tag{8.2}$$

Figure 8.6 show the resulting histogram for all $1,313$ sensors. While there are some poorly functioning sensors (in fact, 31 sensors have no unflagged data at all), the vast majority are performing very well. The median "percent good" from all sensors is 99.1%. Table 8.2 shows a rough breakdown of the "percent good" for the sensors.

Table 8.2: Data quality summary for the $1,313$ considered sensors.

Range of "percent good"	Number of sensors
0	31
$1-50$	28
$51-75$	35
$75-90$	124
$90-95$	85
$95-99$	296
> 99	714

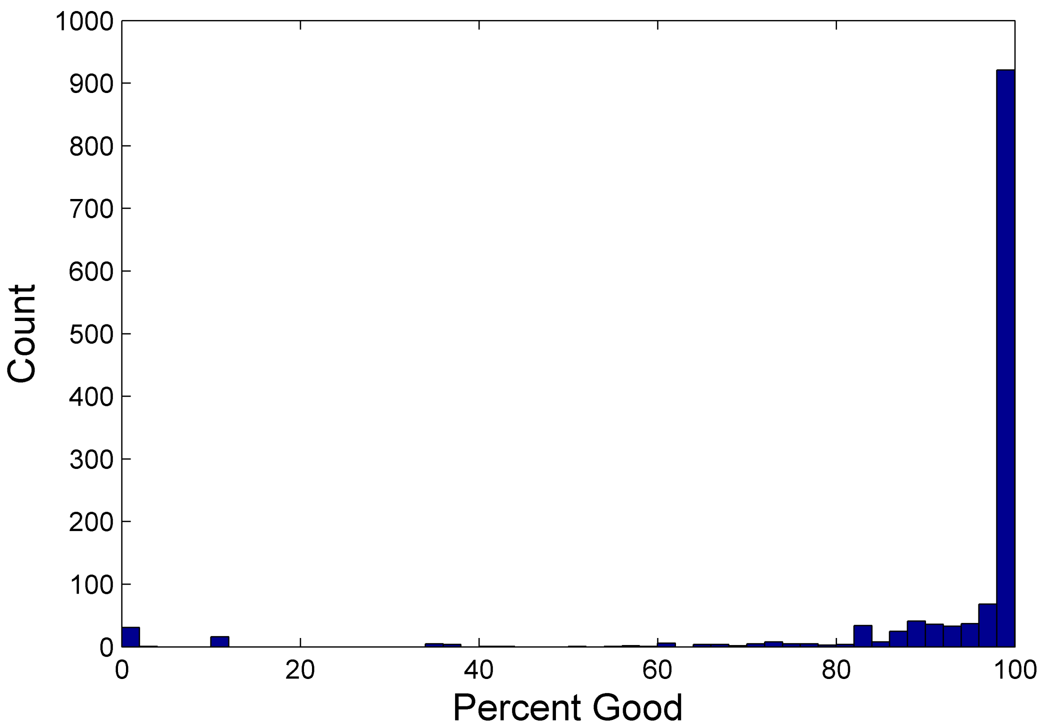

Figure 8.6: A histogram of the "percent good" for all $1,313$ sensors. The "percent good" for a sensor is defined as the number of unflagged data records divided by the number of active data records.

8.3 Sensor-by-sensor Summary Statistics

The following sequence of tables (Table 8.3 through Table 8.74) presents a sensor-by-sensor summary of the filtered data. Each Cell has one or more associated tables, each sensor within the cell has an associated row.

The caption for each table takes the form "Cell # Summary Statistics (`type`, `loc`, `count`)", where:

- `type` is the pavement type
  - `HMA` hot mix asphalt,
  - `PCC` Portland cement concrete, and
  - `UNK` unknown.
- `loc` is the cell location
  - `ML` Mainline, and
  - `LVR` Low Volume Road
- `count` is the total number of data records for this sensor, include missing data.

The "group" is the depth group, as designated in the original database. The reported "number active" is the number of data that are not designated as "missing". The reported "% good" is the percentage of the active data that are unflagged. The six reported statistics for temperatures (median, min, max, median daily range, and max daily range) are all computed using only the unflagged data.

Table 8.3: Cell #1 Summary Statistics (HMA, ML, 395936).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	395926	392729	99.2%	12.0	-30.2	57.5	12.8	36.5
2	2	395903	393729	99.5%	12.1	-29.1	54.4	11.1	32.3
3	5	395936	393640	99.4%	12.4	-22.9	45.3	5.2	16.7
4	6	329094	327103	99.4%	14.0	-19.1	46.9	4.7	14.1
5	7	329093	326570	99.2%	14.7	-16.4	45.0	3.6	10.3
6	8	329094	326255	99.1%	13.3	-13.0	38.5	1.9	6.6
7	9	329094	326177	99.1%	12.8	-10.5	35.5	1.1	3.2
8	10	329094	327374	99.5%	12.1	-5.9	31.9	0.7	2.1
9	11	329094	327457	99.5%	11.9	-2.8	29.1	0.5	2.0
10	12	329094	327205	99.4%	11.7	-1.3	27.0	0.5	4.2
11	13	329094	327097	99.4%	11.3	0.8	22.9	0.8	6.1

Table 8.4: Cell #2 Summary Statistics (HMA, ML, 397173).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	397143	394278	99.3%	12.1	-30.7	55.9	13.3	34.1
2	2	397173	395125	99.5%	12.2	-29.8	53.5	11.6	30.6
3	5	397173	394782	99.4%	12.5	-23.2	44.0	5.8	17.5
4	6	330149	327881	99.3%	14.0	-19.4	49.1	6.4	18.4
5	7	330149	327923	99.3%	13.6	-15.1	42.1	3.6	11.7
6	8	330149	327087	99.1%	13.7	-11.7	38.6	2.0	7.2
7	9	330149	327395	99.2%	12.9	-9.6	35.8	1.2	4.4
8	10	330149	328218	99.4%	12.3	-5.3	32.0	0.7	5.2
9	11	330149	327926	99.3%	12.2	-2.5	29.6	0.5	2.2
10	12	330149	323757	98.1%	11.6	-1.2	27.3	0.6	3.3
11	13	330149	328136	99.4%	11.5	1.1	23.6	0.6	5.5

Table 8.5: Cell #3 Summary Statistics (HMA, ML, 395496).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	346816	342353	98.7%	12.3	-29.9	54.3	13.2	32.6
2	2	346842	343122	98.9%	12.5	-28.6	52.3	11.3	29.4
3	4	395496	393082	99.4%	12.4	-24.1	46.3	6.3	21.9
4	5	395407	370213	93.6%	13.2	-21.2	55.6	5.7	27.9
5	7	329417	327232	99.3%	13.0	-16.2	41.3	3.2	9.5
6	8	329417	326528	99.1%	13.1	-12.5	38.1	1.7	6.1
7	9	329417	326621	99.2%	12.7	-10.3	35.6	0.9	3.4
8	10	329417	327506	99.4%	12.2	-6.3	32.2	0.4	2.3
9	11	329417	327294	99.4%	12.1	-3.2	30.1	0.4	2.5
10	12	329417	327538	99.4%	11.7	-1.2	27.5	0.3	2.3
11	13	280763	277493	98.8%	11.0	1.2	23.0	0.3	2.3

Table 8.6: Cell #4 Summary Statistics (HMA, ML, 396173).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	68334	65811	96.3%	10.5	-30.6	55.4	15.4	33.2
2	2	73722	69026	93.6%	11.5	-29.0	51.7	13.1	32.8
3	4	396169	393991	99.5%	12.0	-24.1	46.2	6.8	19.8
4	6	396165	375475	94.8%	13.0	-19.5	40.5	3.5	21.8
5	8	329417	326670	99.2%	12.8	-11.5	37.7	1.8	6.3
6	9	329419	327158	99.3%	12.5	-8.0	34.9	0.8	2.8
7	10	329419	327553	99.4%	12.1	-4.1	31.5	0.6	2.4
8	11	329419	327447	99.4%	11.8	-2.1	28.7	0.5	2.4
9	12	329419	327485	99.4%	11.6	-1.2	26.8	0.5	2.6
10	13	329419	327681	99.5%	11.7	1.7	22.6	0.5	2.6
11	1	43919	40702	92.7%	5.0	-30.1	54.2	14.6	39.1
12	2	395497	393340	99.5%	11.9	-29.0	53.7	11.1	30.5
13	4	395497	393206	99.4%	12.2	-24.2	46.9	7.0	23.2
14	6	395497	392668	99.3%	12.3	-19.2	40.0	3.1	14.3
15	7	329419	327396	99.4%	12.7	-11.4	37.6	1.8	6.5
16	8	329419	326177	99.0%	12.5	-7.7	34.4	0.9	2.8
17	10	329419	327631	99.5%	12.2	-3.9	31.2	0.6	2.4
18	10	329419	327670	99.5%	11.8	-2.0	28.3	0.6	2.3
19	11	329419	327628	99.5%	11.6	-0.8	26.3	0.4	2.3
20	13	329419	326431	99.1%	11.5	1.5	22.6	0.5	3.9
21	1	229944	227274	98.8%	11.7	-30.2	53.7	13.1	30.0
22	3	229982	227348	98.9%	12.0	-27.6	48.9	9.0	22.9
23	4	396172	391814	98.9%	12.2	-23.7	45.5	6.3	23.9
24	6	396173	391777	98.9%	12.2	-17.6	40.6	2.7	9.4
25	8	396173	392852	99.2%	12.0	-14.4	36.7	1.2	4.6
26	9	329420	327438	99.4%	12.4	-7.7	33.8	0.5	2.3
27	10	329420	327607	99.4%	12.2	-3.4	30.7	0.3	2.2
28	11	329420	327616	99.5%	11.6	-1.6	27.7	0.3	2.0
29	12	329420	326076	99.0%	11.5	-0.5	25.8	0.3	3.7
30	13	329420	327521	99.4%	11.1	1.8	21.8	0.2	2.1

Table 8.7: Cell #4 Summary Statistics (HMA, ML, 396173) continued.

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
31	2	396173	173006	43.7%	14.5	-12.3	35.4	1.9	6.4
32	4	396169	332221	83.9%	6.4	-24.8	50.4	11.2	37.5
33	6	396173	391824	98.9%	11.1	-19.3	42.2	4.7	20.0
34	8	396060	392513	99.1%	11.2	-15.5	36.4	1.9	6.4
35	9	329420	327526	99.4%	11.3	-7.6	33.0	0.7	2.8
36	10	329420	327618	99.5%	11.1	-4.3	29.6	0.3	2.3
37	11	329420	327637	99.5%	10.9	-2.0	26.8	0.3	2.1
38	12	329358	303883	92.3%	10.3	-0.5	24.9	0.3	2.6
39	13	280704	254002	90.5%	10.1	1.7	21.1	0.2	2.6
40	1	347456	320075	92.1%	11.4	-30.6	56.0	13.9	35.4
41	2	347456	320698	92.3%	11.5	-29.2	52.8	11.6	31.3
42	4	347454	320570	92.3%	11.8	-23.9	46.8	7.1	23.4
43	6	396110	370334	93.5%	11.8	-19.3	41.1	3.4	23.1
44	8	329358	303696	92.2%	11.9	-11.6	37.8	1.7	12.8
45	9	329358	304082	92.3%	12.0	-8.1	35.6	0.7	10.3
46	10	329358	304289	92.4%	11.6	-4.0	31.4	0.6	6.0
47	11	329358	304322	92.4%	11.3	-2.0	28.9	0.4	4.1
48	12	329358	304197	92.4%	11.1	-0.8	26.5	0.3	3.5
49	13	329358	303586	92.2%	11.4	1.2	23.5	0.5	6.7
50	1	166585	138695	83.3%	10.0	-30.6	56.0	14.2	35.6
51	2	168428	140735	83.6%	9.7	-29.0	52.3	11.3	64.0
52	4	63871	60607	94.9%	13.1	-23.9	45.1	8.9	24.7
53	6	168754	138521	82.1%	10.5	-20.1	37.7	4.3	45.9
54	7	396173	387829	97.9%	13.6	-16.6	40.4	3.0	8.5
55	9	329420	325871	98.9%	13.4	-9.4	36.9	1.5	5.8
56	10	329420	327177	99.3%	12.3	-4.8	31.9	0.4	3.2
57	11	329420	314360	95.4%	12.6	-2.9	29.5	1.0	10.6
58	12	329420	293775	89.2%	12.1	-1.4	27.2	1.1	4.9
59	13	329420	326078	99.0%	11.4	0.8	23.0	1.1	4.8
60	1	396171	393144	99.2%	11.9	-30.4	57.2	13.5	33.4

Table 8.8: Cell #4 Summary Statistics (HMA, ML, 396173) continued.

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
61	2	396171	394020	99.5%	12.0	-28.9	53.2	11.5	28.8
62	4	396172	393884	99.4%	12.3	-23.4	46.7	7.2	20.1
63	6	396172	393460	99.3%	12.4	-19.8	41.2	3.5	10.7
64	8	329420	326811	99.2%	12.6	-11.7	37.8	1.6	5.8
65	9	329420	327495	99.4%	12.5	-8.0	34.6	0.6	2.5
66	10	329420	327568	99.4%	12.1	-3.7	31.2	0.3	2.2
67	11	329420	327628	99.5%	11.7	-1.8	28.5	0.3	2.1
68	12	329420	327569	99.4%	11.5	-0.6	26.4	0.3	2.1
69	13	329420	327653	99.5%	11.0	1.8	21.6	0.2	2.1

Table 8.9: Cell #5 Summary Statistics (PCC, ML, 396830).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	396830	393418	99.1%	11.6	-30.5	53.7	13.4	30.2
2	2	396822	393945	99.3%	11.8	-29.4	52.0	11.6	27.5
3	3	396830	394686	99.5%	11.9	-27.0	48.5	8.8	23.5
4	6	272936	238186	87.3%	13.0	-21.6	38.7	4.4	21.2
5	6	396830	394939	99.5%	12.0	-20.0	41.0	3.8	13.8
6	7	330183	327831	99.3%	12.6	-15.1	39.1	2.8	8.2
7	8	330183	327397	99.2%	12.2	-12.9	36.0	1.6	5.7
8	9	330183	327183	99.1%	12.1	-10.8	34.1	1.0	2.9
9	10	330183	328186	99.4%	11.7	-6.7	31.2	0.6	1.8
10	11	330183	328723	99.6%	11.6	-3.6	28.6	0.5	1.5
11	12	330183	328730	99.6%	11.3	-1.8	26.3	0.5	1.0
12	13	330183	328746	99.6%	10.8	1.1	21.6	0.4	0.9

Table 8.10: Cell #6 Summary Statistics (PCC, ML, 394665).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	394662	350487	88.8%	11.2	-30.8	53.6	13.0	47.1
2	3	394661	382692	97.0%	10.9	-28.1	47.3	9.8	33.6
3	6	394662	361487	91.6%	9.7	-23.5	44.6	4.8	27.2
4	6	394662	388933	98.5%	11.5	-21.5	38.7	3.8	16.5
5	7	328781	322942	98.2%	12.2	-15.9	36.7	2.8	8.4
6	8	328782	323083	98.3%	12.3	-12.5	33.6	1.4	5.3
7	9	328782	322738	98.2%	12.0	-9.8	31.8	0.7	2.7
8	10	328782	323900	98.5%	11.6	-5.2	30.3	0.5	1.7
9	11	328782	323346	98.3%	11.4	-2.5	28.0	0.4	3.5
10	12	328782	322240	98.0%	11.0	-1.1	25.9	0.4	2.9
11	13	328782	286533	87.1%	10.8	1.6	21.2	0.4	13.3
12	1	394662	387545	98.2%	11.6	-30.5	54.1	13.6	47.4
13	3	394663	389026	98.6%	11.9	-28.1	49.8	9.4	24.3
14	6	386376	379733	98.3%	12.0	-21.3	39.9	4.2	12.0
15	6	393910	380570	96.6%	11.9	-21.4	39.6	3.8	26.2
16	6	394663	388245	98.4%	12.0	-19.1	37.0	2.7	13.1
17	8	328784	272131	82.8%	13.0	-12.0	33.3	1.6	22.9
18	9	328784	272138	82.8%	12.8	-9.2	31.8	0.8	21.3
19	10	328784	273374	83.1%	12.4	-4.7	30.1	0.5	11.9
20	11	328784	273280	83.1%	12.2	-2.6	27.7	0.4	12.0
21	12	328784	273377	83.1%	12.1	-1.2	25.7	0.4	12.0
22	13	328784	273383	83.1%	11.7	2.1	21.2	0.4	11.9
23	1	394665	337790	85.6%	11.0	-31.2	56.9	15.4	47.4
24	3	394665	339076	85.9%	12.0	-26.6	47.4	7.3	37.4
25	5	394665	339960	86.1%	12.2	-24.3	43.9	5.7	33.7
26	6	394665	339788	86.1%	12.2	-20.7	41.2	4.4	32.6
27	7	328784	271787	82.7%	12.9	-16.9	38.3	3.4	29.1
28	8	328784	271363	82.5%	13.0	-13.4	34.2	1.8	24.0
29	9	328784	271663	82.6%	12.8	-10.3	32.2	0.9	21.5
30	10	328784	272849	83.0%	12.4	-5.3	29.6	0.5	21.3

Table 8.11: Cell #6 Summary Statistics (PCC, ML, 394665) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
31	11	246421	212082	86.1%	11.8	-2.7	27.8	0.4	11.9
32	12	328784	270658	82.3%	11.9	-1.4	25.5	0.4	12.0
33	13	303709	234282	77.1%	11.2	1.8	20.8	0.4	14.0
34	5	250198	192421	76.9%	4.7	-24.8	44.8	12.2	35.7
35	6	369589	272138	73.6%	6.4	-22.0	44.1	7.7	35.8
36	7	303709	222452	73.2%	7.6	-20.0	39.3	6.2	26.2
37	8	303709	214620	70.7%	7.7	-19.3	36.6	3.4	30.0
38	9	303709	247673	81.5%	10.4	-14.0	32.7	1.7	11.4
39	10	303710	269388	88.7%	11.6	-7.5	28.9	0.7	12.0
40	11	303710	271633	89.4%	11.2	-5.0	27.1	0.5	12.0
41	12	303710	249324	82.1%	11.3	-3.1	26.0	0.4	11.9
42	13	303710	252979	83.3%	11.3	-1.5	22.0	0.4	10.0
43	1	369589	298564	80.8%	12.4	-30.7	54.1	12.6	49.2
44	3	369591	336576	91.1%	12.4	-28.2	50.5	9.9	42.4
45	6	369591	334901	90.6%	12.6	-23.4	45.8	5.1	24.5
46	7	369591	334833	90.6%	12.8	-21.2	38.3	3.7	20.6
47	7	303710	270639	89.1%	13.3	-14.8	37.0	2.8	19.2
48	8	303710	269664	88.8%	13.2	-12.0	34.6	1.5	12.0
49	9	328784	271849	82.7%	12.7	-9.1	31.7	0.7	21.6
50	10	328784	273277	83.1%	12.4	-4.8	30.1	0.4	11.9
51	11	328784	273274	83.1%	12.2	-2.7	27.8	0.3	11.9
52	12	328784	273282	83.1%	11.9	-1.4	25.6	0.3	11.9
53	13	328784	273286	83.1%	11.4	1.9	21.1	0.3	11.8
54	1	394665	338951	85.9%	12.0	-30.7	54.3	13.2	48.6
55	3	394664	341399	86.5%	11.9	-28.3	49.2	8.8	41.4
56	6	394665	337963	85.6%	12.3	-22.9	40.4	4.4	31.2
57	6	394665	334268	84.7%	14.8	-21.8	45.5	4.7	30.2
58	7	328784	272094	82.8%	12.9	-15.0	36.2	3.0	27.1
59	8	328784	271748	82.7%	12.8	-12.0	33.2	1.5	23.0
60	9	328784	267803	81.5%	13.0	-9.3	31.7	0.8	21.6

Table 8.12: Cell #6 Summary Statistics (PCC, ML, 394665) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
61	10	328784	273280	83.1%	12.4	-5.0	30.2	0.4	11.9
62	11	328784	273365	83.1%	12.2	-2.7	28.0	0.3	11.9
63	12	328784	273281	83.1%	12.0	-1.4	25.8	0.3	11.8
64	13	328784	273279	83.1%	11.3	1.8	20.9	0.3	11.7
65	1	391474	378260	96.6%	12.0	-30.6	53.6	13.9	47.4
66	3	394188	385996	97.9%	12.0	-28.6	49.4	10.2	24.7
67	6	394189	385300	97.7%	12.1	-23.0	40.6	4.6	12.8
68	6	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
69	7	328308	319603	97.3%	12.6	-15.0	36.5	2.8	11.7
70	8	299916	291511	97.2%	12.1	-11.7	33.4	1.4	11.6
71	9	328308	319511	97.3%	12.3	-10.8	31.8	0.7	11.5
72	10	328308	320561	97.6%	11.9	-5.9	30.1	0.5	9.2
73	11	328308	316843	96.5%	11.9	-4.8	28.3	0.4	7.5
74	12	328308	320701	97.7%	11.3	-3.7	25.6	0.4	6.5
75	13	328308	320573	97.6%	10.8	1.6	21.0	0.4	3.1

Table 8.13: Cell #7 Summary Statistics (PCC, ML, 396692).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	395037	376799	95.4%	10.7	-31.5	53.2	13.1	48.3
2	3	395039	378398	95.8%	11.1	-29.3	49.0	9.4	39.3
3	6	395041	377542	95.6%	11.2	-26.6	45.4	6.8	31.9
4	6	395040	377947	95.7%	11.2	-24.3	43.2	5.6	28.9
5	7	328348	311453	94.9%	11.8	-12.1	36.6	2.0	16.0
6	8	328348	311227	94.8%	12.0	-8.7	34.7	1.0	13.2
7	9	328346	311245	94.8%	11.6	-6.9	32.6	0.6	11.8
8	10	328344	312172	95.1%	11.4	-3.4	29.5	0.3	7.4
9	11	328343	312087	95.0%	11.2	-1.6	27.1	0.4	3.7
10	12	328343	310974	94.7%	11.0	-0.4	25.2	0.3	3.8
11	13	328344	312182	95.1%	10.6	2.3	20.5	0.3	3.8
12	1	395051	377259	95.5%	11.2	-31.1	54.5	14.1	48.3
13	3	395053	378583	95.8%	11.5	-29.5	50.6	10.5	40.3
14	5	395049	373923	94.7%	11.4	-25.5	46.2	7.1	32.6
15	6	395049	377932	95.7%	11.6	-23.7	43.4	5.7	28.6
16	7	328356	311402	94.8%	11.8	-12.6	36.7	2.2	16.0
17	8	330002	327209	99.2%	12.7	-9.2	36.3	1.2	4.7
18	9	330002	327110	99.1%	12.4	-7.0	32.9	0.6	2.3
19	10	330002	328261	99.5%	12.0	-3.4	29.7	0.3	1.6
20	11	330002	328276	99.5%	11.5	-1.8	27.1	0.3	1.0
21	12	330002	328264	99.5%	11.5	-0.5	25.3	0.3	1.1
22	13	330002	328265	99.5%	11.1	2.4	20.8	0.2	0.8
23	1	396692	392721	99.0%	11.8	-29.8	53.1	13.5	30.7
24	3	396692	393386	99.2%	12.1	-26.4	46.8	8.0	21.7
25	5	396691	394438	99.4%	12.3	-23.8	45.1	5.7	16.4
26	6	378818	376328	99.3%	11.5	-19.6	41.0	3.7	10.9
27	7	330002	326868	99.1%	12.7	-14.5	38.7	2.6	10.2
28	8	330002	326851	99.0%	12.7	-11.5	36.2	1.3	4.9
29	9	330002	326977	99.1%	12.3	-8.5	33.7	0.6	5.9
30	10	330002	328178	99.4%	11.7	-4.3	30.1	0.3	1.9

Table 8.14: Cell #7 Summary Statistics (PCC, ML, 396692) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
31	11	330002	327935	99.4%	11.5	-2.4	27.2	0.4	2.6
32	12	330002	328240	99.5%	11.1	-0.9	25.1	0.3	1.2
33	13	330002	328300	99.5%	10.5	2.1	19.8	0.3	0.8
34	5	396692	383657	96.7%	11.0	-18.3	40.5	3.9	14.7
35	6	396692	387948	97.8%	10.5	-16.7	33.9	2.6	9.1
36	7	330002	326855	99.0%	10.6	-10.9	32.4	2.0	6.9
37	8	330002	327145	99.1%	10.7	-7.7	30.1	0.9	4.2
38	9	330002	327503	99.2%	10.8	-5.7	28.3	0.5	2.1
39	10	330002	328193	99.5%	10.1	-3.4	25.5	0.3	2.1
40	11	330002	327880	99.4%	10.1	-1.6	23.5	0.3	1.9
41	12	330002	328129	99.4%	9.9	-0.2	21.9	0.3	1.0
42	13	330002	325155	98.5%	9.6	2.1	17.9	0.3	1.2
43	1	396691	393373	99.2%	12.0	-31.4	54.6	14.1	31.6
44	3	396692	394504	99.4%	12.2	-29.0	49.8	10.0	24.8
45	6	396692	393801	99.3%	12.3	-22.9	42.0	5.1	13.8
46	6	330002	328048	99.4%	12.6	-17.4	41.1	4.5	12.3
47	7	330002	327316	99.2%	12.6	-12.1	37.6	2.5	7.2
48	8	330002	326978	99.1%	12.4	-8.6	34.1	1.0	5.6
49	9	312825	281616	90.0%	11.6	-6.1	28.9	0.6	25.1
50	10	312690	282416	90.3%	11.2	-3.1	26.4	0.3	23.1
51	11	312637	282327	90.3%	10.8	-1.6	24.5	0.3	23.2
52	12	312482	282327	90.3%	10.6	-0.6	22.8	0.3	23.2
53	13	311602	282421	90.6%	10.5	2.3	19.0	0.2	20.2
54	1	379706	347769	91.6%	11.4	-31.0	53.4	13.9	58.1
55	3	379508	348679	91.9%	11.3	-29.6	49.4	10.7	51.3
56	5	379490	348665	91.9%	11.3	-25.6	44.8	7.6	46.1
57	6	379361	348474	91.9%	11.5	-21.3	39.4	4.7	35.7
58	7	312416	281794	90.2%	11.8	-12.0	35.1	2.5	30.2
59	8	312758	281612	90.0%	11.9	-9.0	31.6	1.2	28.1
60	9	312834	281804	90.1%	11.7	-6.6	29.5	0.6	26.7

Table 8.15: Cell #7 Summary Statistics (PCC, ML, 396692) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
61	10	312779	282326	90.3%	11.4	-3.3	26.8	0.3	24.0
62	11	312639	282327	90.3%	10.9	-1.8	24.7	0.3	22.8
63	12	312269	282327	90.4%	10.8	-0.5	23.2	0.3	21.5
64	13	293208	282270	96.3%	10.4	2.1	19.0	0.2	0.8
65	1	396163	392335	99.0%	12.0	-31.8	54.5	14.5	32.6
66	3	396550	393250	99.2%	12.2	-29.8	50.6	11.1	27.6
67	5	396692	392716	99.0%	12.3	-27.2	46.4	7.9	24.0
68	6	396692	392428	98.9%	12.2	-23.6	42.1	5.6	21.8
69	7	330002	326314	98.9%	12.8	-11.1	36.7	2.0	14.2
70	8	330002	325586	98.7%	12.8	-9.0	35.6	1.3	7.2
71	9	330002	325625	98.7%	12.4	-7.1	33.1	0.7	6.2
72	10	329999	325931	98.8%	11.9	-3.6	30.0	0.4	8.1
73	11	329999	325195	98.5%	11.5	-1.8	27.5	0.3	9.4
74	12	330001	247509	75.0%	12.7	-0.6	26.0	0.7	7.9
75	13	330002	323770	98.1%	10.7	2.2	20.8	0.2	5.9

Table 8.16: Cell #8 Summary Statistics (PCC, ML, 397696).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	397696	394248	99.1%	11.8	-31.7	52.9	13.9	30.5
2	2	397696	394875	99.3%	12.0	-30.6	50.8	12.0	27.5
3	3	397696	395358	99.4%	12.1	-28.1	46.6	8.6	21.4
4	6	397696	394706	99.2%	12.2	-23.9	42.2	5.7	14.8
5	6	330330	328451	99.4%	12.7	-16.5	39.6	4.1	10.7
6	7	330330	328109	99.3%	12.8	-11.4	36.0	2.0	6.2
7	8	330330	327869	99.3%	12.6	-8.7	34.4	1.2	4.1
8	9	330330	328153	99.3%	12.3	-6.5	32.2	0.7	2.1
9	10	120853	119317	98.7%	12.0	-3.2	25.9	0.5	1.3
10	11	330322	328708	99.5%	11.7	-1.6	26.8	0.4	0.9
11	12	330322	328716	99.5%	11.4	-0.5	25.0	0.4	0.9
12	13	330322	328647	99.5%	11.0	2.3	20.5	0.4	1.3

Table 8.17: Cell #9 Summary Statistics (PCC, ML, 397518).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	397511	394218	99.2%	11.7	-31.0	53.5	13.2	29.9
2	2	397513	392946	98.9%	12.7	-30.0	55.1	11.4	26.8
3	3	397512	395272	99.4%	12.1	-28.1	48.2	8.8	22.1
4	6	397513	394555	99.3%	12.0	-23.9	43.3	5.7	18.0
5	7	328802	313592	95.4%	13.3	-14.4	38.8	3.2	17.5
6	7	330126	327482	99.2%	12.6	-11.5	36.6	1.9	9.8
7	8	330126	327522	99.2%	12.3	-9.4	34.6	1.1	8.4
8	9	330034	327301	99.2%	12.1	-6.7	32.4	0.6	5.8
9	10	330044	328196	99.4%	11.8	-3.3	29.4	0.4	5.9
10	11	330123	328109	99.4%	11.5	-1.7	26.8	0.4	6.6
11	12	330125	328188	99.4%	11.3	-0.6	25.1	0.4	5.3
12	13	330053	328173	99.4%	10.8	2.4	20.1	0.3	3.4

Table 8.18: Cell #10 Summary Statistics (PCC, ML, 397608).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	395162	384143	97.2%	11.4	-31.8	52.3	12.2	28.5
2	3	397605	388796	97.8%	12.0	-29.3	48.0	8.6	22.4
3	5	390867	388396	99.4%	12.3	-24.1	44.2	6.0	16.3
4	6	390867	388232	99.3%	12.3	-21.0	40.1	3.8	10.6
5	7	390867	387380	99.1%	12.4	-16.2	37.1	2.1	6.9
6	8	323803	320574	99.0%	12.9	-8.8	34.2	1.0	3.6
7	9	323803	320975	99.1%	12.5	-6.4	31.9	0.6	2.0
8	10	323803	321717	99.4%	12.0	-3.2	29.0	0.4	1.2
9	11	323803	321631	99.3%	12.0	-1.6	26.6	0.4	2.7
10	12	323803	321642	99.3%	11.6	-0.4	24.7	0.4	1.9
11	13	323803	321652	99.3%	11.1	2.3	20.3	0.4	1.6
12	1	390867	387558	99.2%	12.1	-31.7	54.1	13.3	30.4
13	3	390867	388563	99.4%	12.4	-29.6	50.1	9.8	24.5
14	5	390863	388397	99.4%	12.6	-24.6	45.7	6.6	17.7
15	6	390867	388190	99.3%	12.5	-19.3	39.4	3.4	9.7
16	7	390867	387216	99.1%	12.5	-16.3	37.1	2.1	6.8
17	8	330544	328346	99.3%	12.4	-8.7	34.4	1.0	3.4
18	9	330544	328375	99.3%	12.3	-6.2	32.4	0.6	3.3
19	10	330544	328927	99.5%	11.9	-3.0	29.6	0.5	1.9
20	11	330544	329030	99.5%	11.6	-1.5	26.9	0.4	1.1
21	12	330544	329040	99.5%	11.5	-0.2	25.1	0.4	1.0
22	13	330544	317339	96.0%	11.9	2.4	21.5	0.5	2.1
23	1	397608	393431	98.9%	11.6	-30.5	53.0	12.7	29.9
24	3	397608	394355	99.2%	11.9	-27.2	48.6	8.3	22.3
25	5	397608	395828	99.6%	12.1	-23.6	44.8	5.6	15.8
26	6	397608	391954	98.6%	11.8	-22.7	48.4	6.0	20.7
27	7	397608	390481	98.2%	12.2	-20.7	43.9	4.3	14.9
28	8	330544	326466	98.8%	12.7	-14.8	38.8	2.4	7.7
29	9	330544	327091	99.0%	12.5	-11.6	36.3	1.3	4.6
30	10	330544	328392	99.3%	12.1	-6.5	31.7	0.5	3.7

Table 8.19: Cell #10 Summary Statistics (PCC, ML, 397608).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
31	11	330544	329030	99.5%	11.7	-3.1	29.1	0.5	1.5
32	12	330544	317163	96.0%	12.1	-1.7	26.4	0.6	2.8
33	13	330544	328319	99.3%	10.9	1.6	21.1	0.5	2.9
34	6	397604	394458	99.2%	10.4	-20.4	39.6	3.8	15.6
35	7	397608	393261	98.9%	10.6	-18.6	36.8	2.7	10.4
36	8	330544	327278	99.0%	11.0	-11.7	34.9	2.1	7.0
37	9	330544	327998	99.2%	10.9	-8.7	32.3	1.2	4.5
38	9	330544	327736	99.2%	10.9	-6.5	30.4	0.8	2.4
39	10	330544	328957	99.5%	10.5	-3.9	27.0	0.5	4.0
40	11	330544	328957	99.5%	10.0	-2.5	24.1	0.5	2.1
41	12	330544	329052	99.5%	10.2	-0.8	22.6	0.5	1.0
42	13	330544	328982	99.5%	10.0	1.7	18.6	0.5	1.5

Table 8.20: Cell #11 Summary Statistics (PCC, ML, 397620).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	397620	394358	99.2%	11.8	-29.8	52.6	13.0	29.8
2	3	397619	395845	99.6%	12.0	-27.5	48.6	9.5	23.5
3	5	397619	395858	99.6%	12.2	-23.6	43.5	5.6	15.4
4	6	397619	395590	99.5%	12.2	-20.8	40.3	3.5	9.9
5	7	397619	393448	99.0%	12.4	-19.1	39.1	2.8	7.9
6	8	330046	326610	99.0%	12.8	-11.9	36.5	1.5	5.1
7	9	330046	327148	99.1%	12.3	-9.5	34.0	0.8	2.6
8	10	330046	328521	99.5%	11.9	-5.1	30.7	0.5	1.9
9	11	330046	327264	99.2%	11.8	-2.8	28.1	0.5	3.8
10	12	245259	212026	86.4%	12.9	-0.4	23.9	0.4	13.4
11	13	236385	230582	97.5%	12.5	1.5	19.9	0.4	2.6

Table 8.21: Cell #12 Summary Statistics (PCC, ML, 398043).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	398040	394224	99.0%	11.1	-29.9	50.0	11.9	27.3
2	3	398040	396244	99.5%	11.5	-27.6	46.5	8.5	21.3
3	5	398040	396458	99.6%	11.8	-23.9	42.7	5.3	14.6
4	6	398040	396094	99.5%	11.9	-21.0	39.7	3.4	9.6
5	7	398040	393853	98.9%	12.2	-19.9	39.2	2.9	8.4
6	8	330504	327760	99.2%	12.6	-12.6	36.6	1.6	5.6
7	9	330504	327750	99.2%	12.2	-10.1	34.1	0.8	2.6
8	10	330504	328726	99.5%	11.8	-5.3	30.6	0.4	2.0
9	11	330504	329020	99.6%	11.6	-2.6	28.2	0.3	3.1
10	12	330504	329025	99.6%	11.4	-1.2	25.9	0.3	1.8
11	13	330504	328876	99.5%	11.0	1.7	21.1	0.3	1.1
12	1	398040	395063	99.3%	11.8	-30.1	52.2	13.3	29.6
13	3	398040	396376	99.6%	12.0	-27.9	48.3	9.7	23.6
14	5	398040	396396	99.6%	12.2	-23.4	43.8	6.1	16.2
15	6	398040	395919	99.5%	12.1	-20.6	39.5	3.4	9.5
16	7	398040	393650	98.9%	12.3	-19.7	38.7	3.0	8.3
17	8	330507	327670	99.1%	12.4	-12.4	36.0	1.5	5.4
18	9	330507	327846	99.2%	12.3	-9.7	33.8	0.8	2.6
19	10	330507	328920	99.5%	11.8	-4.9	30.4	0.4	1.8
20	11	330507	328992	99.5%	11.6	-2.6	27.8	0.3	1.5
21	12	330507	328992	99.5%	11.5	-1.0	25.7	0.3	2.1
22	13	330507	329073	99.6%	11.2	2.0	21.3	0.2	0.9
23	1	398043	394044	99.0%	11.2	-30.0	56.5	13.8	38.9
24	3	345583	342030	99.0%	10.9	-26.9	47.3	8.3	23.4
25	5	398043	395942	99.5%	12.0	-23.9	43.8	6.0	17.2
26	6	398043	366610	92.1%	9.8	-25.1	48.5	7.9	29.9
27	7	398043	368255	92.5%	10.7	-21.1	43.5	5.1	24.4
28	8	313145	302034	96.5%	11.1	-16.7	39.8	2.8	10.6
29	9	330507	326481	98.8%	12.6	-12.9	36.3	1.4	5.8
30	10	330507	325392	98.5%	12.3	-7.7	31.7	0.5	5.4

Table 8.22: Cell #12 Summary Statistics (PCC, ML, 398043) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
31	11	330507	327760	99.2%	12.2	-4.1	28.9	0.6	3.2
32	12	313133	311408	99.4%	10.9	-1.9	26.5	0.4	2.7
33	13	304240	302698	99.5%	10.9	1.4	21.5	0.3	2.0
34	5	398043	383222	96.3%	11.0	-20.9	39.3	4.5	18.1
35	6	398043	389499	97.9%	10.7	-19.5	35.6	3.1	11.7
36	7	330507	326765	98.9%	11.0	-12.0	33.6	2.2	7.7
37	8	330507	327317	99.0%	11.0	-9.1	31.4	1.2	4.4
38	9	330507	326474	98.8%	11.0	-6.9	33.1	0.7	6.3
39	10	330507	322118	97.5%	10.9	-4.3	26.7	0.5	6.2
40	11	330507	310004	93.8%	11.1	-1.9	24.4	0.5	7.6
41	12	250954	249612	99.5%	10.2	-0.3	21.9	0.3	2.3
42	13	315	0	0.0%	NaN	NaN	NaN	3.6	7.8

Table 8.23: Cell #13 Summary Statistics (PCC, ML, 397360).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	397360	394241	99.2%	11.9	-30.5	53.5	13.9	31.2
2	3	397360	395643	99.6%	12.2	-27.6	48.7	9.6	23.2
3	5	397360	395485	99.5%	12.3	-23.5	44.4	6.4	16.3
4	6	397357	395044	99.4%	12.3	-20.3	39.9	3.4	9.4
5	7	397360	393049	98.9%	12.6	-18.0	38.8	2.7	8.1
6	8	329696	327116	99.2%	12.9	-11.3	36.0	1.4	4.8
7	9	329696	326887	99.1%	12.4	-8.9	33.6	0.7	4.5
8	10	329696	328075	99.5%	12.1	-4.5	30.5	0.4	6.0
9	11	329696	327974	99.5%	11.8	-2.6	28.0	0.4	7.6
10	12	329696	327664	99.4%	11.6	-1.2	26.0	0.3	1.9
11	13	329696	328034	99.5%	11.1	1.7	21.3	0.3	1.3

Table 8.24: Cell #14 Summary Statistics (HMA, ML, 391092).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	391092	384822	98.4%	11.8	-30.2	56.4	13.8	33.6
2	2	391092	385555	98.6%	11.9	-28.9	53.7	11.9	29.7
3	4	391092	385255	98.5%	12.1	-24.2	46.6	7.6	19.9
4	7	391086	384203	98.2%	12.2	-18.5	39.8	3.0	11.0
5	8	391092	383838	98.1%	12.4	-15.8	38.2	2.0	24.5
6	9	323881	317689	98.1%	12.7	-8.0	34.9	0.7	19.1
7	10	323881	318345	98.3%	11.9	-3.9	31.0	0.4	14.3
8	11	323881	318067	98.2%	11.7	-2.1	28.6	0.4	8.0
9	12	323881	316173	97.6%	11.7	-1.0	26.9	0.4	5.0
10	13	323881	318234	98.3%	11.2	1.7	21.7	0.3	3.5

Table 8.25: Cell #15 Summary Statistics (HMA, ML, 391094).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	391092	384799	98.4%	11.3	-31.3	56.0	13.3	33.8
2	2	391091	385710	98.6%	11.4	-30.2	53.2	11.4	29.8
3	4	391092	385368	98.5%	11.7	-24.7	46.0	6.7	18.4
4	7	391092	385057	98.5%	12.0	-17.4	39.4	2.5	10.2
5	8	323881	317848	98.1%	12.3	-11.4	37.3	1.6	22.3
6	9	323881	317611	98.1%	12.1	-7.7	34.1	0.6	18.0
7	10	323883	318364	98.3%	11.9	-3.8	31.1	0.3	14.8
8	11	323883	318376	98.3%	11.6	-1.9	28.4	0.3	11.9
9	12	323883	318346	98.3%	11.6	-0.7	26.5	0.3	10.2
10	13	323883	318326	98.3%	11.1	2.1	21.4	0.2	5.2
11	1	391093	387812	99.2%	11.7	-30.5	56.2	12.7	35.5
12	2	391094	388826	99.4%	12.0	-29.0	53.4	10.8	30.6
13	4	391094	385417	98.5%	11.9	-24.5	47.0	7.1	19.4
14	7	391067	384769	98.4%	12.1	-17.7	39.2	2.5	7.9
15	8	323883	317088	97.9%	12.4	-11.5	37.4	1.6	6.0
16	9	323883	319813	98.7%	12.1	-7.8	34.1	0.6	7.4
17	10	323883	318380	98.3%	11.9	-3.8	30.8	0.3	14.3
18	11	323883	318283	98.3%	11.6	-2.0	28.2	0.3	12.0
19	12	323883	318357	98.3%	11.6	-0.7	26.2	0.3	9.7
20	13	323883	318346	98.3%	11.1	2.0	21.3	0.2	4.8
21	1	391094	387802	99.2%	11.3	-29.6	53.3	12.2	29.1
22	2	391094	388019	99.2%	11.4	-26.5	51.4	9.9	25.1
23	4	391094	387328	99.0%	11.9	-23.1	44.8	5.8	16.8
24	7	391094	387997	99.2%	11.9	-17.8	42.7	3.9	11.9
25	8	323883	321857	99.4%	12.1	-13.9	38.6	2.2	7.0
26	9	323883	322086	99.4%	11.9	-9.4	34.6	0.8	2.9
27	10	323883	322568	99.6%	11.6	-4.8	30.8	0.4	1.6
28	11	323883	322593	99.6%	11.5	-2.0	28.3	0.3	1.3
29	12	323883	322521	99.6%	11.1	-0.9	25.7	0.3	2.5
30	13	323883	322622	99.6%	10.6	1.9	20.7	0.3	3.6

Table 8.26: Cell #15 Summary Statistics (HMA, ML, 391094) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
31	7	323883	321908	99.4%	10.3	-12.7	36.9	2.6	8.7
32	8	323883	321976	99.4%	10.3	-10.0	33.4	1.4	5.2
33	9	323883	322663	99.6%	10.1	-6.9	30.0	0.8	2.4
34	10	323883	322585	99.6%	10.0	-3.4	26.6	0.3	2.1
35	11	323883	322574	99.6%	9.8	-1.7	24.2	0.4	2.5
36	12	323883	321443	99.2%	9.8	-0.5	22.4	0.4	2.2
37	13	323883	319396	98.6%	9.7	2.0	18.4	0.3	2.3

Table 8.27: Cell #16 Summary Statistics (HMA, ML, 361429).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	360421	358117	99.4%	11.6	-30.2	55.3	14.2	34.4
2	3	361428	359698	99.5%	11.7	-27.8	49.4	9.9	26.2
3	4	361428	359396	99.4%	11.9	-25.3	46.5	7.4	20.7
4	6	361429	357635	99.0%	11.9	-20.2	41.7	3.9	12.0
5	7	293902	291463	99.2%	12.5	-15.5	40.7	3.0	9.0
6	8	293902	291309	99.1%	12.6	-12.9	38.0	1.6	7.2
7	9	293902	290659	98.9%	12.3	-11.0	35.7	0.9	9.2
8	10	293902	290773	98.9%	12.0	-6.9	32.3	0.5	6.6
9	11	293252	291547	99.4%	12.0	-3.7	30.0	0.4	1.9
10	12	293252	291656	99.5%	11.6	-2.1	27.8	0.4	1.8
11	13	293252	290648	99.1%	10.9	0.8	22.6	0.4	0.8

Table 8.28: Cell #17 Summary Statistics (HMA, ML, 361429).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	360779	358783	99.4%	11.0	-30.7	54.9	14.3	33.8
2	2	360779	359152	99.5%	11.2	-30.8	51.2	11.7	29.2
3	4	360779	358673	99.4%	11.5	-25.3	46.4	7.3	19.9
4	6	360779	357501	99.1%	11.8	-20.3	41.9	3.9	11.8
5	7	293252	291073	99.3%	12.1	-15.5	40.1	2.8	8.3
6	8	293252	290963	99.2%	12.5	-13.0	37.4	1.5	5.1
7	9	293252	290574	99.1%	12.5	-10.6	35.5	0.9	3.0
8	10	293252	291517	99.4%	12.2	-6.3	32.3	0.5	1.8
9	11	293252	291544	99.4%	12.2	-3.2	29.8	0.4	2.9
10	12	293252	291639	99.4%	12.1	-1.4	27.6	0.4	0.8
11	13	293252	291666	99.5%	11.6	1.6	22.9	0.3	1.5
12	1	360779	358616	99.4%	11.9	-30.9	56.3	14.8	35.0
13	2	360779	359188	99.6%	12.0	-29.3	53.2	12.4	30.7
14	4	360779	358817	99.5%	12.3	-24.2	47.3	7.8	20.9
15	6	360779	358363	99.3%	12.4	-20.1	41.5	3.6	10.6
16	7	293252	291238	99.3%	12.7	-14.9	40.0	2.7	7.9
17	8	112118	110407	98.5%	10.4	-10.9	32.3	1.5	4.6
18	9	293252	290586	99.1%	12.6	-10.4	35.4	0.9	3.0
19	10	293252	291529	99.4%	12.3	-6.3	32.1	0.5	2.3
20	11	293252	291629	99.4%	12.2	-3.2	29.6	0.4	1.5
21	12	293252	291621	99.4%	11.9	-1.5	27.3	0.4	1.0
22	13	293252	291541	99.4%	11.3	1.5	22.3	0.3	1.0
23	1	360779	357847	99.2%	11.4	-30.0	52.5	12.8	29.3
24	2	360779	359123	99.5%	11.6	-28.4	50.6	10.8	25.8
25	4	360778	357633	99.1%	12.0	-23.7	45.5	6.5	18.1
26	6	360779	357552	99.1%	12.2	-19.0	43.1	4.5	13.0
27	7	293252	291199	99.3%	12.7	-15.3	40.1	2.9	8.2
28	8	293252	290801	99.2%	12.4	-13.0	36.9	1.6	5.4
29	9	293252	290873	99.2%	12.3	-10.9	34.9	0.9	3.1
30	10	293252	291262	99.3%	12.1	-6.6	32.0	0.5	4.7

Table 8.29: Cell #17 Summary Statistics (HMA, ML, 361429) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
31	11	293252	291559	99.4%	11.9	-3.3	29.0	0.3	1.8
32	12	293252	291475	99.4%	11.7	-1.5	26.6	0.3	2.9
33	13	293252	291568	99.4%	11.0	1.6	21.5	0.3	0.9
34	2	360779	297645	82.5%	12.3	-22.6	39.3	4.7	14.1
35	4	360779	279778	77.5%	9.3	-21.1	49.7	5.7	24.4
36	6	360779	356771	98.9%	10.9	-19.4	39.4	3.4	12.5
37	7	293252	290632	99.1%	11.2	-11.8	34.4	2.2	7.6
38	8	293252	290613	99.1%	11.1	-9.4	33.7	1.2	5.0
39	9	293252	291233	99.3%	11.1	-7.7	34.0	0.7	3.2
40	10	120050	118170	98.4%	11.0	-4.3	25.7	0.3	1.9
41	11	293252	291502	99.4%	10.8	-2.5	26.0	0.3	1.0
42	12	293252	291480	99.4%	10.6	-1.0	24.0	0.3	3.3
43	13	293252	290727	99.1%	10.4	1.3	20.1	1.2	4.1
44	1	360779	358313	99.3%	11.6	-30.7	55.2	13.4	33.0
45	2	360779	359191	99.6%	11.6	-29.1	51.6	11.2	28.6
46	4	360779	358864	99.5%	12.1	-23.9	45.5	6.7	19.1
47	6	360779	357835	99.2%	12.3	-21.3	42.3	4.0	12.4
48	7	120050	118222	98.5%	12.7	-13.9	38.1	3.6	9.9
49	8	293252	290850	99.2%	12.6	-12.8	37.9	1.6	5.8
50	9	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
51	10	293311	283337	96.6%	13.0	-6.5	32.9	1.2	12.2
52	11	293252	285654	97.4%	12.6	-4.0	30.0	1.0	6.1
53	12	108479	72509	66.8%	12.8	-0.1	25.6	1.3	10.5
54	13	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
55	1	432	0	0.0%	NaN	NaN	NaN	6.5	12.0
56	2	558	0	0.0%	NaN	NaN	NaN	7.1	9.4
57	4	604	0	0.0%	NaN	NaN	NaN	4.3	5.1
58	6	360779	343693	95.3%	13.7	-18.9	41.8	4.0	13.5
59	7	293252	291239	99.3%	13.0	-15.1	40.6	3.0	9.0
60	8	293252	291204	99.3%	12.6	-13.0	37.4	1.7	5.8

Table 8.30: Cell #17 Summary Statistics (HMA, ML, 361429) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
61	9	293252	290610	99.1%	12.6	-10.8	35.5	1.0	3.3
62	10	293252	259714	88.6%	12.1	-6.5	29.1	0.6	8.7
63	11	293252	291647	99.5%	12.2	-3.1	29.8	0.5	1.8
64	12	293252	291655	99.5%	11.9	-1.5	27.4	0.4	1.2
65	13	293252	291685	99.5%	11.2	1.4	22.2	0.4	0.9
66	1	360779	358508	99.4%	11.6	-31.0	56.0	14.3	34.1
67	2	360779	359158	99.6%	11.7	-29.9	52.7	11.8	30.0
68	4	360779	358697	99.4%	12.0	-24.2	45.6	6.9	19.6
69	6	360779	357763	99.2%	12.2	-21.7	41.8	4.0	12.3
70	7	293902	291740	99.3%	12.7	-15.1	40.7	2.9	17.6
71	8	293902	291556	99.2%	12.7	-12.7	37.7	1.6	16.4
72	9	293902	290528	98.9%	12.6	-10.5	35.6	0.9	15.3
73	10	293902	291907	99.3%	12.3	-6.5	32.2	0.5	4.2
74	11	293902	291657	99.2%	12.3	-3.1	30.0	0.4	8.3
75	12	293902	291289	99.1%	12.2	-1.3	27.6	0.4	5.5
76	13	293902	291314	99.1%	11.3	1.5	22.2	0.3	4.3

Table 8.31: Cell #18 Summary Statistics (HMA, ML, 362362).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	3	14897	8810	59.1%	-3.2	-26.0	18.3	10.0	42.3
2	7	362362	359624	99.2%	12.7	-18.5	42.0	4.0	12.3
3	11	295318	293943	99.5%	12.0	-2.3	28.7	0.3	2.5
4	6	362360	357751	98.7%	13.4	-20.5	44.5	4.9	14.4
5	9	111633	62547	56.0%	15.8	-7.1	35.7	2.1	17.2
6	4	95516	80095	83.9%	8.3	-23.7	42.2	6.6	40.1
7	12	295318	267833	90.7%	13.3	-1.1	27.8	0.4	10.2
8	10	295318	294050	99.6%	12.0	-5.2	31.2	0.4	2.6
9	13	295318	262626	88.9%	13.9	1.0	24.6	0.7	6.3
10	8	295318	292643	99.1%	12.9	-12.0	37.7	1.9	7.7
11	1	8118	7342	90.4%	-4.8	-28.2	23.7	13.1	24.0

Table 8.32: Cell #19 Summary Statistics (HMA, ML, 362353).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	362351	360696	99.5%	12.0	-27.5	56.8	15.0	35.3
2	3	362351	361055	99.6%	12.3	-24.8	50.1	10.3	26.2
3	4	362351	361151	99.7%	12.4	-22.8	45.7	7.1	19.6
4	6	362351	360220	99.4%	12.5	-19.8	42.4	4.4	13.6
5	7	362351	359872	99.3%	12.6	-17.2	40.2	3.1	9.5
6	8	295320	292958	99.2%	12.8	-12.9	37.3	1.7	5.9
7	9	295320	292747	99.1%	12.7	-10.6	35.3	1.0	3.4
8	10	295320	293281	99.3%	12.4	-6.6	32.0	0.4	1.8
9	11	295320	293504	99.4%	12.0	-3.3	29.4	0.4	2.7
10	12	295320	293745	99.5%	11.9	-1.7	27.1	0.3	2.5
11	13	295320	293758	99.5%	11.2	1.4	22.3	0.2	2.1

Table 8.33: Cell #20 Summary Statistics (HMA, ML, 361470).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	361470	359498	99.5%	12.2	-27.0	55.2	12.1	32.8
2	3	361469	359886	99.6%	12.3	-24.3	48.6	8.7	24.3
3	4	361468	359572	99.5%	12.5	-22.3	44.3	6.0	18.2
4	6	361464	359265	99.4%	12.5	-20.4	41.9	3.9	15.8
5	7	294940	293246	99.4%	13.0	-14.7	39.6	2.8	9.9
6	8	294940	292303	99.1%	13.2	-12.3	37.1	1.6	5.6
7	9	294940	292440	99.2%	12.7	-10.3	35.0	0.9	5.9
8	10	294940	293500	99.5%	12.4	-6.2	31.9	0.4	3.3
9	11	294940	293512	99.5%	12.1	-3.0	29.4	0.4	3.1
10	12	294940	293443	99.5%	11.9	-1.4	27.1	0.3	2.9
11	13	294940	293577	99.5%	11.3	1.7	22.3	0.3	2.9

Table 8.34: Cell #21 Summary Statistics (HMA, ML, 361459).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	361456	359666	99.5%	12.1	-27.1	55.8	13.7	33.4
2	2	361454	360162	99.6%	12.2	-26.0	52.5	11.4	28.9
3	4	361455	360122	99.6%	12.4	-23.3	45.8	6.9	19.4
4	6	361455	359125	99.4%	12.6	-21.0	42.1	4.2	12.8
5	7	294940	293068	99.4%	12.9	-16.2	40.9	3.4	10.2
6	8	294943	292053	99.0%	13.0	-13.1	37.7	1.9	6.4
7	9	294943	290733	98.6%	12.8	-10.4	35.5	1.1	4.0
8	10	294943	293123	99.4%	12.4	-5.8	31.9	0.5	3.2
9	11	294943	293378	99.5%	12.1	-2.7	29.3	0.3	3.1
10	12	294943	293144	99.4%	12.0	-1.2	27.2	0.3	3.0
11	13	294943	292950	99.3%	11.5	1.7	22.6	0.3	3.0

Table 8.35: Cell #22 Summary Statistics (HMA, ML, 359304).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	359301	356869	99.3%	11.5	-28.3	55.4	14.0	33.7
2	2	359301	357636	99.5%	11.7	-26.9	51.7	11.2	28.1
3	4	359301	357424	99.5%	11.9	-24.3	46.0	7.2	19.7
4	6	359301	356204	99.1%	12.1	-21.2	42.0	4.4	13.9
5	7	292726	289937	99.0%	12.5	-16.6	40.9	3.4	13.4
6	8	292726	289347	98.8%	12.7	-12.0	38.1	1.7	8.8
7	9	292726	289755	99.0%	12.4	-7.9	35.2	0.8	6.0
8	10	292726	290900	99.4%	12.1	-3.7	31.7	0.4	1.6
9	11	292726	290973	99.4%	11.8	-2.2	29.3	0.3	1.1
10	12	292726	290347	99.2%	11.7	-1.0	27.6	0.3	2.5
11	13	292726	290995	99.4%	11.3	0.8	23.9	0.3	1.6
12	1	359298	357035	99.4%	12.0	-27.9	56.0	13.6	34.1
13	2	359300	357568	99.5%	12.1	-27.1	53.7	12.2	31.2
14	4	359300	357359	99.5%	12.3	-24.0	46.2	7.3	20.9
15	6	359301	356291	99.2%	12.4	-21.7	43.1	5.0	15.2
16	7	292726	290033	99.1%	12.6	-17.8	42.1	4.1	12.6
17	8	292729	289348	98.8%	12.7	-12.7	38.6	2.0	8.1
18	9	292729	289782	99.0%	12.6	-8.8	35.3	0.9	4.7
19	10	292729	290916	99.4%	12.2	-4.0	31.8	0.4	2.5
20	11	292729	290948	99.4%	11.7	-2.5	29.2	0.3	1.2
21	12	292729	290950	99.4%	11.7	-1.3	27.4	0.3	1.8
22	13	292729	290595	99.3%	11.5	1.0	24.1	0.3	0.8
23	1	359302	356261	99.2%	11.4	-28.0	53.8	13.8	31.8
24	2	359302	357128	99.4%	11.6	-26.9	50.7	11.3	27.7
25	4	359304	356772	99.3%	11.9	-23.1	45.4	6.5	18.6
26	6	359304	356170	99.1%	12.0	-20.5	43.9	5.1	14.8
27	7	292729	290178	99.1%	12.3	-17.4	40.5	3.3	9.5
28	8	292729	290033	99.1%	12.3	-13.0	36.9	1.7	6.9
29	9	292729	290324	99.2%	12.3	-9.2	34.4	0.8	3.6
30	10	292729	290927	99.4%	11.9	-4.4	31.0	0.4	2.1

Table 8.36: Cell #22 Summary Statistics (HMA, ML, 359304) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
31	11	292729	290913	99.4%	11.5	-2.5	28.4	0.3	1.0
32	12	292729	290986	99.4%	11.3	-1.3	26.5	0.3	0.9
33	13	292729	290962	99.4%	11.1	0.6	23.3	0.3	2.6
34	2	359304	340953	94.9%	11.3	-23.8	45.9	7.8	21.8
35	5	359302	354622	98.7%	11.2	-20.3	43.0	4.6	14.2
36	6	359302	353488	98.4%	10.8	-17.7	37.9	3.2	12.3
37	7	292729	288981	98.7%	11.2	-11.8	35.5	2.0	10.4
38	8	292729	289061	98.7%	11.9	-8.2	34.1	1.0	7.5
39	9	292729	290882	99.4%	11.1	-6.1	31.0	0.5	2.2
40	10	292729	290958	99.4%	10.9	-3.3	28.0	0.3	3.4
41	11	292729	290874	99.4%	10.9	-1.7	25.8	0.3	1.4
42	12	292729	290901	99.4%	10.7	-0.7	24.4	0.3	2.7
43	13	292729	291002	99.4%	10.6	1.2	21.4	0.2	2.1
44	1	359298	357068	99.4%	11.9	-27.5	55.9	14.3	34.6
45	2	359298	357571	99.5%	12.1	-26.5	53.4	12.4	31.0
46	4	359298	357504	99.5%	12.3	-23.6	46.7	7.7	21.3
47	6	359304	356243	99.1%	12.4	-21.2	42.5	5.0	15.0
48	7	292729	289964	99.1%	12.6	-17.0	41.0	3.8	11.7
49	8	292729	288885	98.7%	12.6	-12.6	38.2	2.0	8.0
50	9	292729	289961	99.1%	12.6	-8.6	35.3	0.9	4.4
51	10	292729	290663	99.3%	12.2	-3.8	30.9	0.4	2.1
52	11	292729	290893	99.4%	11.8	-1.9	28.1	0.3	1.0
53	12	292729	290911	99.4%	11.7	-0.6	26.1	0.3	1.4
54	13	292729	290891	99.4%	11.5	1.9	22.2	0.2	1.7
55	1	359297	356977	99.4%	11.9	-27.7	55.9	14.4	34.7
56	2	359298	357665	99.5%	12.0	-26.7	53.0	12.3	30.7
57	4	359297	357498	99.5%	12.3	-23.8	46.5	7.7	21.3
58	6	359304	356315	99.2%	12.4	-20.8	41.7	4.6	13.7
59	7	292729	289974	99.1%	12.6	-17.2	40.9	3.8	11.2
60	8	292729	289067	98.7%	12.6	-12.3	37.8	1.9	8.3

Table 8.37: Cell #22 Summary Statistics (HMA, ML, 359304) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
61	9	292729	290107	99.1%	12.5	-8.4	35.1	0.8	4.7
62	10	292729	290728	99.3%	12.1	-3.7	30.6	0.3	1.5
63	11	292729	290969	99.4%	11.7	-1.7	27.8	0.3	0.9
64	12	292729	290979	99.4%	11.4	-0.5	25.6	0.3	0.8
65	13	292729	290893	99.4%	11.1	1.8	21.5	0.2	2.5
66	1	359297	356791	99.3%	11.8	-28.2	57.0	15.1	36.3
67	2	359298	357578	99.5%	11.9	-27.4	54.1	13.2	32.8
68	4	359298	357481	99.5%	12.2	-24.2	47.4	8.0	22.1
69	6	359304	356245	99.1%	12.5	-20.7	42.9	4.8	14.4
70	7	292729	290145	99.1%	13.0	-16.3	41.2	3.3	9.9
71	8	292729	289137	98.8%	12.7	-11.9	38.6	1.6	7.0
72	9	292729	289849	99.0%	12.5	-7.9	35.6	0.7	3.1
73	10	292729	290952	99.4%	12.2	-3.7	31.3	0.3	1.9
74	11	292729	290892	99.4%	11.8	-2.1	28.7	0.3	2.3
75	12	292729	290985	99.4%	11.6	-0.9	26.9	0.3	1.0
76	13	292729	290986	99.4%	11.2	1.3	23.0	0.3	0.8

Table 8.38: Cell #23 Summary Statistics (HMA, ML, 397337).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	397337	394917	99.4%	12.0	-27.9	56.4	13.0	34.4
2	2	397337	395812	99.6%	12.1	-26.9	53.0	11.1	30.1
3	6	397337	392557	98.8%	12.4	-23.8	45.8	7.4	20.4
4	6	397337	394616	99.3%	12.4	-20.0	40.5	4.2	11.7
5	7	330552	328184	99.3%	12.8	-12.7	39.0	2.7	8.6
6	8	330552	327137	99.0%	12.9	-9.1	35.6	1.3	5.7
7	9	330552	327614	99.1%	12.6	-7.0	33.2	0.7	2.8
8	10	330552	328666	99.4%	12.2	-3.2	29.9	0.4	1.3
9	11	330552	328781	99.5%	11.9	-1.4	25.5	0.3	1.2
10	12	330552	328720	99.4%	11.8	-0.9	27.2	0.3	1.0
11	13	330552	329090	99.6%	11.4	1.8	21.5	0.5	1.5

Table 8.39: Cell #24 Summary Statistics (HMA, LVR, 361050).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	361050	359989	99.7%	12.1	-30.9	55.7	15.1	34.4
2	3	361050	357012	98.9%	12.7	-27.6	50.5	10.6	26.7
3	5	361050	356477	98.7%	13.0	-22.6	46.4	6.9	19.4
4	7	293209	286684	97.8%	13.6	-17.5	40.6	3.3	9.1
5	8	293209	289037	98.6%	13.1	-14.1	37.3	1.8	6.2
6	9	293209	290512	99.1%	13.0	-11.0	35.1	1.0	3.4
7	10	293209	291559	99.4%	12.8	-5.8	31.5	0.4	2.2
8	11	293209	291941	99.6%	12.5	-2.2	28.6	0.4	2.3
9	12	293209	291906	99.6%	12.3	0.0	26.4	0.3	1.0
10	13	293209	291855	99.5%	11.8	2.8	22.4	0.3	0.9

Table 8.40: Cell #25 Summary Statistics (HMA, LVR, 361055).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	361055	359793	99.7%	11.9	-29.0	53.7	14.6	31.7
2	2	361055	359291	99.5%	12.1	-27.6	50.4	11.8	27.0
3	5	361055	357054	98.9%	12.7	-20.8	44.5	5.8	15.5
4	7	293214	289143	98.6%	13.1	-15.9	40.9	3.4	9.3
5	8	293214	290503	99.1%	13.2	-11.8	38.2	1.9	6.4
6	9	293214	291149	99.3%	13.3	-7.3	35.4	0.8	2.6
7	10	293214	291907	99.6%	12.7	-3.3	32.2	0.4	3.0
8	11	293214	291386	99.4%	12.8	-1.1	30.0	0.4	1.6
9	12	293214	291788	99.5%	12.7	-0.2	28.0	0.4	1.0
10	13	293214	291718	99.5%	12.0	2.2	23.6	0.3	1.0
11	1	359825	357791	99.4%	12.1	-28.2	56.4	14.7	46.7
12	2	359825	357498	99.4%	12.3	-26.6	52.5	11.9	29.3
13	5	359825	355221	98.7%	12.8	-21.1	45.3	6.4	21.0
14	7	293214	289287	98.7%	13.1	-16.1	41.1	3.7	10.2
15	8	293214	290524	99.1%	13.1	-11.7	37.8	2.0	6.5
16	9	293214	290876	99.2%	12.8	-7.3	34.5	0.8	2.8
17	10	293214	291878	99.5%	12.3	-3.3	31.3	0.5	2.1
18	11	293214	291344	99.4%	12.3	-1.2	29.0	0.4	1.2
19	12	293214	291735	99.5%	12.2	-0.4	27.0	0.3	1.0
20	13	293214	291663	99.5%	11.4	2.0	22.4	0.3	0.9
21	1	361055	352502	97.6%	12.1	-27.5	54.7	13.5	34.0
22	2	361025	356576	98.8%	12.1	-25.9	51.6	11.0	28.9
23	4	361029	351372	97.3%	12.5	-21.0	45.3	6.3	19.2
24	8	361052	358035	99.2%	12.0	-10.6	34.9	1.5	10.7
25	9	293214	291228	99.3%	12.1	-5.5	32.6	0.7	2.5
26	10	293214	291891	99.5%	11.8	-3.2	30.5	0.4	2.4
27	11	293214	291785	99.5%	11.7	-0.8	27.8	0.3	1.1
28	12	293214	291851	99.5%	11.5	0.1	25.9	0.3	0.9
29	12	293214	291791	99.5%	11.4	0.8	24.3	0.3	0.9
30	13	293214	291777	99.5%	10.9	2.9	20.2	0.3	0.9

Table 8.41: Cell #25 Summary Statistics (HMA, LVR, 361055) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
31	6	293214	287529	98.1%	9.1	-9.7	29.0	1.3	5.8
32	7	293214	264650	90.3%	10.3	-7.0	27.1	0.7	2.9
33	9	293214	283233	96.6%	9.7	-3.5	25.4	0.5	1.5
34	10	293214	289945	98.9%	9.6	-0.8	23.5	0.4	1.4
35	11	293214	291757	99.5%	9.7	0.3	21.9	0.3	1.1
36	12	293214	291244	99.3%	9.8	1.0	20.8	0.3	1.2
37	13	293214	291597	99.4%	10.0	2.9	18.2	0.3	1.1

Table 8.42: Cell #26 Summary Statistics (HMA, LVR, 280788).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	149378	147891	99.0%	11.3	-27.7	52.9	14.2	30.7
2	3	149353	147513	98.8%	11.7	-23.5	47.1	9.6	22.5
3	5	159516	155835	97.7%	12.9	-17.9	40.1	4.7	14.7
4	7	112986	111540	98.7%	12.6	-12.0	37.0	2.4	8.1
5	8	112986	110882	98.1%	12.8	-9.5	34.4	1.3	4.7
6	9	112986	111015	98.3%	12.4	-7.2	32.2	0.7	8.6
7	10	112986	111295	98.5%	11.9	-3.1	28.9	0.4	7.8
8	11	112991	112331	99.4%	11.4	-1.1	26.8	0.3	1.1
9	12	112991	106422	94.2%	12.3	0.1	25.2	0.3	4.9
10	13	112991	112288	99.4%	10.8	2.8	21.5	0.2	1.0
11	1	148696	147679	99.3%	11.5	-27.7	55.4	16.0	47.5
12	3	158395	146415	92.4%	11.9	-24.7	49.4	11.7	49.0
13	5	148672	145944	98.2%	12.1	-18.3	42.6	5.7	19.5
14	7	112991	111074	98.3%	12.4	-11.8	40.4	3.2	12.7
15	8	81811	79864	97.6%	11.5	-8.9	34.6	1.5	11.3
16	9	103096	96777	93.9%	10.5	-10.2	32.2	0.9	14.8
17	10	82379	80547	97.8%	10.8	-2.8	28.7	0.3	11.0
18	11	83344	69173	83.0%	13.7	-0.9	26.6	0.4	5.7
19	12	81284	80527	99.1%	10.8	0.6	24.8	0.3	0.7
20	13	87604	80566	92.0%	10.4	3.5	21.0	0.3	5.1
21	1	181067	170357	94.1%	11.5	-24.3	56.0	12.2	35.9
22	2	149943	148190	98.8%	11.1	-22.8	52.2	12.7	46.9
23	3	149361	147686	98.9%	11.4	-17.5	45.9	7.5	21.0
24	7	180299	153424	85.1%	11.2	-13.1	40.8	3.8	56.7
25	8	80523	79192	98.3%	11.5	-9.5	35.8	2.1	6.2
26	9	80523	79442	98.7%	10.8	-6.5	32.9	1.0	3.5
27	10	80523	79460	98.7%	10.6	-2.7	29.0	0.4	1.9
28	11	80523	79460	98.7%	10.5	-0.2	26.0	0.3	0.9
29	12	80523	79487	98.7%	10.3	1.0	24.0	0.3	0.8
30	13	80523	79464	98.7%	9.9	3.5	20.0	0.2	0.7

Table 8.43: Cell #26 Summary Statistics (HMA, LVR, 280788) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
31	8	112228	100517	89.6%	11.6	-2.6	27.7	0.7	3.2
32	9	112228	107842	96.1%	9.8	-1.1	25.6	0.4	1.8
33	9	112991	106207	94.0%	10.0	-0.5	23.9	0.3	7.3
34	10	112991	108915	96.4%	9.3	0.5	21.7	0.3	6.3
35	11	112991	111670	98.8%	8.9	-0.7	20.4	0.2	5.4
36	12	112991	112000	99.1%	8.9	-1.1	19.4	0.2	3.5
37	13	112991	111273	98.5%	9.0	4.4	16.6	0.2	0.9
38	1	99716	99255	99.5%	14.4	-25.0	60.6	19.6	36.9
39	1	99718	99258	99.5%	14.5	-25.0	58.6	17.9	34.2
40	2	99717	99304	99.6%	14.6	-24.9	57.1	16.4	31.9
41	2	52923	52457	99.1%	9.8	-23.4	55.4	13.5	29.9
42	3	99720	99547	99.8%	14.9	-24.3	53.7	13.3	27.1
43	3	99715	99538	99.8%	15.0	-23.6	52.4	12.1	25.2
44	5	99714	99525	99.8%	14.9	-21.9	50.6	10.4	22.4
45	7	99711	91940	92.2%	15.4	-17.1	47.6	9.0	19.8
46	8	99703	66727	66.9%	6.6	-15.8	39.7	6.6	14.6
47	9	99707	97963	98.3%	15.8	-12.2	39.0	3.0	7.6

Table 8.44: Cell #27 Summary Statistics (HMA, LVR, 340418).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	118744	104602	88.1%	9.6	-30.0	54.0	16.5	64.0
2	3	111374	106370	95.5%	10.8	-23.9	50.4	8.7	47.8
3	6	30398	21199	69.7%	8.3	-15.4	39.1	6.8	32.2
4	7	53262	52270	98.1%	13.5	-11.2	36.4	2.9	7.2
5	8	53262	52272	98.1%	13.5	-7.8	33.4	1.4	3.9
6	9	53262	52601	98.8%	13.3	-5.6	31.3	0.6	2.2
7	10	53262	52942	99.4%	12.0	-1.9	28.0	0.3	1.7
8	11	53262	52895	99.3%	11.8	0.3	26.2	0.3	1.1
9	12	53262	52951	99.4%	11.7	1.8	24.7	0.2	0.6
10	13	53262	52953	99.4%	10.5	3.9	21.0	0.2	0.5
11	1	120671	119894	99.4%	11.8	-28.4	53.7	12.6	32.3
12	3	120665	118952	98.6%	12.4	-25.7	47.8	8.2	25.8
13	5	53262	52246	98.1%	14.0	-16.2	43.6	6.4	16.2
14	7	53262	52248	98.1%	13.8	-12.2	38.3	3.5	8.8
15	8	53262	52276	98.1%	13.7	-8.5	34.4	1.6	4.3
16	9	53262	52281	98.2%	13.7	-6.2	31.8	0.7	2.6
17	10	53262	52954	99.4%	12.0	-2.5	28.2	0.3	1.1
18	11	53262	52664	98.9%	11.4	0.1	26.0	0.3	3.6
19	12	53262	49237	92.4%	12.0	1.0	24.1	0.3	3.2
20	13	53262	37335	70.1%	13.8	3.2	20.9	0.5	3.3
21	1	121341	120063	98.9%	11.4	-27.9	55.4	15.5	33.6
22	3	121341	119164	98.2%	11.9	-21.4	48.3	9.4	23.7
23	6	53262	52054	97.7%	13.3	-14.6	41.1	4.9	13.4
24	6	53262	52277	98.2%	13.3	-12.2	38.6	3.7	9.8
25	8	53262	52269	98.1%	13.4	-7.7	34.4	1.7	4.5
26	9	53262	52221	98.0%	13.0	-5.4	31.5	0.8	2.8
27	10	53262	52949	99.4%	11.3	-1.7	27.5	0.3	1.0
28	11	53262	52965	99.4%	10.9	0.5	25.2	0.3	0.7
29	12	53262	52964	99.4%	10.6	1.7	23.3	0.2	0.5
30	13	53262	52962	99.4%	9.8	4.0	20.0	0.2	0.4

Table 8.45: Cell #27 Summary Statistics (HMA, LVR, 340418) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
31	7	53262	51039	95.8%	10.4	-4.0	27.0	0.9	5.2
32	8	53262	51853	97.4%	9.9	-2.2	25.6	0.4	4.7
33	9	53262	52653	98.9%	9.0	-1.0	24.6	0.3	4.0
34	10	53262	52857	99.2%	8.6	0.4	23.1	0.2	2.2
35	11	53262	52962	99.4%	8.6	1.4	20.7	0.2	0.9
36	12	53262	52944	99.4%	9.1	2.4	19.9	0.2	0.6
37	13	53262	52958	99.4%	8.8	4.3	17.5	0.2	0.5
38	1	8628	0	0.0%	NaN	NaN	NaN	123.2	441.3
39	3	219060	218409	99.7%	12.4	-23.3	50.3	9.2	30.4
40	6	219046	193872	88.5%	12.3	-20.4	51.3	10.3	39.0
41	7	219059	217418	99.3%	13.1	-12.8	37.7	2.4	9.0
42	8	219050	217772	99.4%	13.0	-11.8	36.5	1.8	8.8
43	9	219059	218157	99.6%	12.9	-9.9	34.9	1.1	6.1
44	1	8309	7535	90.7%	12.4	0.5	32.8	8.0	54.5
45	3	219049	218474	99.7%	12.6	-20.9	50.7	7.7	29.2
46	6	3410	0	0.0%	NaN	NaN	NaN	24.4	34.9
47	7	219058	217601	99.3%	12.9	-12.4	37.4	2.5	9.5
48	8	219056	217954	99.5%	12.9	-11.3	36.4	2.0	8.1
49	9	219057	218428	99.7%	12.7	-9.3	34.2	1.2	8.0
50	9	219068	218527	99.8%	12.6	-8.0	33.1	0.7	4.8
51	10	219052	218820	99.9%	12.4	-5.8	31.3	0.4	3.9
52	10	219058	218905	99.9%	12.2	-3.8	29.8	0.3	3.3
53	10	219057	218915	99.9%	12.0	-2.6	28.3	0.3	3.6
54	11	219063	218798	99.9%	11.7	-1.7	27.2	0.2	3.7
55	11	219056	218781	99.9%	11.4	-0.9	26.0	0.2	3.3
56	12	219062	218803	99.9%	11.3	-0.3	24.9	0.2	3.5
57	12	219071	218806	99.9%	11.3	0.2	24.1	0.2	3.6
58	13	219071	218824	99.9%	11.2	0.7	23.3	0.2	3.9
59	13	219071	218807	99.9%	11.1	1.3	22.4	0.2	3.4
60	13	219071	218804	99.9%	11.0	1.7	21.6	0.2	3.3

Table 8.46: Cell #27 Summary Statistics (HMA, LVR, 340418) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
61	3	219054	216521	98.8%	11.0	-20.2	45.5	6.6	26.2
62	6	219057	216933	99.0%	11.4	-14.4	37.8	3.2	16.0
63	7	219050	216803	99.0%	11.3	-12.7	34.9	2.7	11.7
64	8	219063	216938	99.0%	11.1	-11.5	33.4	2.3	9.2
65	9	219052	217723	99.4%	11.2	-9.6	32.0	1.8	6.1

Table 8.47: Cell #28 Summary Statistics (HMA, LVR, 358677).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	121157	120720	99.6%	11.5	-27.8	52.7	13.9	30.1
2	3	121157	119839	98.9%	12.1	-22.8	44.9	7.7	18.9
3	6	53788	51436	95.6%	14.0	-16.4	43.0	6.2	15.9
4	7	53788	52803	98.2%	13.3	-13.5	38.3	3.5	9.1
5	8	53788	52813	98.2%	13.3	-10.4	35.2	1.9	4.8
6	9	53788	52809	98.2%	13.5	-7.9	32.7	1.0	2.6
7	10	55639	53497	96.2%	11.8	-3.8	28.8	0.5	9.7
8	11	53788	53481	99.4%	10.9	-1.1	26.1	0.4	0.9
9	12	53788	53490	99.4%	11.0	0.7	24.6	0.4	0.7
10	13	53862	52806	98.0%	10.1	3.6	21.6	0.3	2.9
11	1	216880	215825	99.5%	12.8	-23.2	50.9	10.7	26.3
12	3	216873	216313	99.7%	12.7	-20.3	44.8	6.7	18.1
13	6	216864	215303	99.3%	13.2	-13.2	38.6	3.0	7.8
14	7	216877	216065	99.6%	13.3	-11.0	36.5	1.8	5.7
15	8	216880	216062	99.6%	13.2	-10.0	35.4	1.3	5.0
16	9	216884	216548	99.8%	13.1	-8.3	33.9	0.7	2.9
17	1	216871	212575	98.0%	12.3	-21.0	47.4	8.2	21.4
18	3	216869	216200	99.7%	12.9	-16.8	42.8	5.1	14.0
19	6	216872	214470	98.9%	13.3	-16.3	41.7	4.4	12.0
20	7	216862	216243	99.7%	13.0	-9.6	35.3	1.5	5.1
21	8	216868	216352	99.8%	12.9	-8.8	34.1	1.2	4.0
22	9	216876	216462	99.8%	12.8	-7.1	32.9	0.6	2.0
23	9	216863	216581	99.9%	12.7	-5.6	31.7	0.4	1.8
24	10	216878	216618	99.9%	12.5	-3.6	30.1	0.3	1.5
25	10	216877	216635	99.9%	12.2	-2.4	28.4	0.2	1.2
26	10	216886	216657	99.9%	11.9	-1.3	26.9	0.2	0.9
27	11	216881	216552	99.8%	11.7	-0.7	25.8	0.2	0.7
28	11	216893	216661	99.9%	11.5	-0.1	24.8	0.2	0.6
29	12	901	0	0.0%	NaN	NaN	NaN	29.6	45.5
30	12	216895	216678	99.9%	11.4	1.1	23.1	0.2	0.4

Table 8.48: Cell #28 Summary Statistics (HMA, LVR, 358677).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
31	13	216893	216677	99.9%	11.3	1.4	22.4	0.2	0.4
32	13	216893	216673	99.9%	11.2	1.9	21.7	0.2	0.4
33	13	216893	216667	99.9%	11.1	2.3	21.2	0.2	0.4
34	3	216868	214399	98.9%	11.1	-17.5	43.0	5.9	19.1
35	6	216875	215099	99.2%	11.3	-13.9	36.1	2.9	9.3
36	7	216889	215405	99.3%	11.3	-10.9	32.8	1.7	6.0
37	8	216880	215654	99.4%	11.5	-9.3	31.8	1.5	3.7
38	9	216878	215942	99.6%	11.6	-7.5	31.1	1.6	3.7
39	1	18709	18575	99.3%	25.8	-11.3	58.2	24.2	33.0
40	1	3034	2553	84.1%	9.5	-10.0	37.0	22.5	30.1
41	2	0	0	0.0%	NaN			NaN	
42	3	18709	18670	99.8%	26.5	-7.2	50.1	15.2	22.8

Table 8.49: Cell #29 Summary Statistics (HMA, LVR, 391524).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	391523	388582	99.2%	12.4	-27.6	55.6	13.8	44.9
2	2	391523	388477	99.2%	12.5	-26.1	52.6	11.5	39.6
3	5	391522	384759	98.3%	13.1	-19.2	44.3	5.7	29.0
4	6	324224	316214	97.5%	13.5	-17.9	43.6	5.2	26.8
5	7	324224	319033	98.4%	13.5	-15.3	40.4	3.5	24.9
6	8	324224	320802	98.9%	13.5	-11.3	38.1	1.8	19.2
7	9	324224	321115	99.0%	13.1	-8.7	37.6	0.9	17.7
8	10	324224	321951	99.3%	12.7	-4.3	31.3	0.5	14.0
9	11	324224	322087	99.3%	12.4	-1.7	28.4	0.4	12.0
10	12	324224	321821	99.3%	12.1	-0.4	26.3	0.4	10.1
11	13	324225	322183	99.4%	11.6	2.5	21.7	0.4	6.2

Table 8.50: Cell #30 Summary Statistics (HMA, LVR, 391615).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	391208	362690	92.7%	10.9	-28.7	54.6	12.9	30.6
2	2	391208	361105	92.3%	11.1	-27.1	51.3	10.6	26.5
3	5	323116	304234	94.2%	11.6	-17.2	43.3	4.8	14.9
4	7	323116	303093	93.8%	13.2	-14.1	40.2	2.9	11.8
5	8	323116	287856	89.1%	12.6	-13.4	37.7	1.9	27.2
6	9	323481	285999	88.4%	12.9	-11.7	35.7	1.0	23.4
7	10	323481	284356	87.9%	12.7	-5.3	32.3	0.5	18.0
8	11	323480	284437	87.9%	12.5	-2.9	29.7	0.5	14.9
9	12	323481	284330	87.9%	12.3	-1.6	27.7	0.4	13.0
10	13	323482	286262	88.5%	11.9	-0.8	23.1	0.4	10.6
11	1	391573	375266	95.8%	11.8	-27.5	55.5	13.8	33.7
12	2	391573	366822	93.7%	11.5	-26.5	52.3	11.4	31.0
13	4	391572	359555	91.8%	12.5	-22.3	45.1	6.1	27.8
14	7	323482	316602	97.9%	12.9	-14.4	41.3	3.3	24.6
15	8	323482	305608	94.5%	12.8	-10.9	38.2	1.8	21.9
16	9	323480	300514	92.9%	13.3	-8.6	35.8	0.9	21.7
17	10	323481	284474	87.9%	12.6	-5.7	32.1	0.5	20.3
18	11	323481	284565	88.0%	12.4	-3.8	29.3	0.4	17.2
19	12	323482	284448	87.9%	12.1	-2.2	27.2	0.4	11.2
20	13	323481	286164	88.5%	11.5	-0.8	22.2	0.4	12.2
21	1	391573	374981	95.8%	11.2	-28.6	58.6	14.1	35.8
22	2	391576	386003	98.6%	11.8	-27.4	53.0	12.2	30.4
23	4	391576	382887	97.8%	12.6	-21.1	44.1	5.7	16.7
24	7	391576	386004	98.6%	12.5	-14.8	41.0	3.4	10.3
25	8	323484	320592	99.1%	12.5	-11.0	37.9	1.9	6.0
26	9	323484	321109	99.3%	12.4	-8.6	37.4	1.0	3.4
27	10	323484	321740	99.5%	12.1	-4.0	31.1	0.5	3.2
28	11	323484	321717	99.5%	11.6	-2.5	28.0	0.5	3.2
29	12	323484	266884	82.5%	13.7	-1.3	25.8	0.7	3.7
30	13	323484	320886	99.2%	11.0	1.6	21.1	0.4	3.5

Table 8.51: Cell #30 Summary Statistics (HMA, LVR, 391615) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
31	7	323484	299451	92.6%	10.1	-7.8	27.5	1.0	3.3
32	8	323484	296942	91.8%	10.3	-5.8	26.1	0.7	2.7
33	9	323484	311414	96.3%	9.4	-3.9	25.1	0.5	2.4
34	10	323484	320735	99.2%	9.4	-1.6	23.2	0.4	2.4
35	11	323484	321884	99.5%	9.4	-0.5	21.7	0.4	2.3
36	12	323484	321639	99.4%	9.6	0.3	20.6	0.4	3.0
37	13	323484	321728	99.5%	9.7	2.4	17.8	0.4	3.3

Table 8.52: Cell #31 Summary Statistics (HMA, LVR, 491186).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	399299	303685	76.1%	11.9	-28.3	57.9	11.0	43.4
2	3	194593	192571	99.0%	13.0	-24.7	46.5	8.0	23.4
3	5	332117	239075	72.0%	12.8	-21.3	46.0	7.3	36.6
4	7	332118	237321	71.5%	13.2	-15.7	40.5	3.6	20.8
5	8	332118	238209	71.7%	13.2	-12.3	38.0	2.0	12.2
6	9	332118	238825	71.9%	13.2	-9.7	36.0	1.1	10.3
7	10	332118	240754	72.5%	12.7	-5.2	32.3	0.4	8.4
8	11	332118	240644	72.5%	12.2	-2.7	29.4	0.3	7.1
9	12	332118	240735	72.5%	12.4	-1.0	27.5	0.3	5.5
10	13	122406	120702	98.6%	11.0	2.2	20.6	0.3	2.5
11	1	96858	72574	74.9%	20.7	-21.2	59.5	18.6	33.4
12	1	96867	87358	90.2%	16.7	-24.4	56.8	16.1	30.9
13	1	96863	87545	90.4%	16.8	-23.5	56.1	15.2	30.0
14	2	96854	69977	72.2%	22.0	-9.9	53.0	11.9	26.9
15	2	96859	90225	93.2%	16.1	-22.5	53.0	11.9	27.6
16	3	96852	72953	75.3%	22.6	-14.6	52.0	11.1	26.1
17	3	96840	67102	69.3%	24.1	-8.1	49.1	8.5	21.8
18	3	96837	40405	41.7%	14.2	-11.0	43.2	4.5	13.1
19	5	96847	72719	75.1%	19.6	-9.4	41.0	3.4	10.9
20	7	96839	72951	75.3%	21.9	-7.3	39.4	2.6	8.8
21	8	96855	81587	84.2%	19.3	-12.3	38.6	2.1	7.5
22	9	96855	75636	78.1%	20.5	-9.4	36.3	1.2	6.5
23	10	96853	64211	66.3%	21.4	-2.9	34.4	0.9	6.4
24	10	96861	64216	66.3%	20.5	-2.8	32.8	0.9	6.4
25	11	96861	58268	60.2%	21.3	-2.6	31.2	0.8	6.2
26	11	96855	55582	57.4%	21.7	-1.4	30.0	0.8	6.6

Table 8.53: Cell #32 Summary Statistics (UNK, LVR, 549485).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	6	331554	119307	36.0%	10.2	-13.4	35.3	2.3	10.8
2	7	331554	120360	36.3%	9.9	-13.4	35.2	2.3	10.5
3	8	331554	117821	35.5%	10.4	-10.4	32.4	1.2	5.1
4	9	331554	120936	36.5%	9.6	-7.9	29.7	0.6	2.5
5	10	331554	120252	36.3%	9.7	-5.8	27.9	0.5	1.9
6	10	331554	121068	36.5%	9.9	-2.6	25.8	0.3	1.4
7	11	331554	118940	35.9%	10.0	-0.9	23.4	0.3	1.0
8	12	331554	118217	35.7%	9.8	0.4	21.8	0.3	1.2
9	13	331554	117991	35.6%	9.9	2.6	18.8	0.2	0.9
10	1	234713	25675	10.9%	17.7	-22.9	47.5	13.2	27.7
11	1	234713	25772	11.0%	17.7	-22.2	46.7	12.4	26.0
12	1	234713	25587	10.9%	17.9	-21.7	45.3	11.3	24.5
13	2	234713	25231	10.7%	18.7	-15.8	35.2	4.0	17.5
14	3	234713	25134	10.7%	18.7	-19.4	42.2	8.1	19.2
15	4	234713	25231	10.7%	18.8	-18.6	40.8	7.1	16.9
16	6	234713	25688	10.9%	18.1	-20.5	43.7	8.8	22.7
17	8	234713	25138	10.7%	18.5	-12.5	32.3	2.1	13.6
18	1	234713	25577	10.9%	18.6	-22.6	47.6	12.5	26.1
19	1	234712	25583	10.9%	19.0	-22.3	47.2	12.2	25.6
20	1	234713	25682	10.9%	18.7	-21.5	45.9	11.1	23.8
21	2	234713	25227	10.7%	19.1	-20.5	44.4	9.4	21.1
22	3	234063	25232	10.8%	19.2	-19.7	42.2	7.9	18.5
23	4	234713	25236	10.8%	19.3	-19.0	40.8	6.9	16.1
24	6	234713	25237	10.8%	19.5	-15.5	35.6	3.7	8.9
25	8	234063	25140	10.7%	19.3	-11.8	33.1	2.1	5.9

Table 8.54: Cell #32 Summary Statistics (UNK, LVR, 549485) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
26	1	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
27	1	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
28	1	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
29	2	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
30	2	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
31	3	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
32	3	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
33	4	0	0	0.0%	NaN	NaN	NaN	NaN	NaN

Table 8.55: Cell #33 Summary Statistics (HMA, LVR, 305023).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	6	53491	52388	97.9%	10.5	-9.9	36.3	4.7	13.9
2	7	53491	53306	99.7%	9.9	-6.7	30.9	2.1	5.8
3	8	53491	53193	99.4%	9.6	-4.0	27.9	1.0	2.5
4	10	53491	53397	99.8%	8.6	-1.1	24.1	0.3	1.4
5	11	53491	53392	99.8%	8.5	0.5	21.8	0.3	1.1
6	12	53491	53376	99.8%	8.7	1.5	20.3	0.3	1.0
7	13	53491	53383	99.8%	8.2	3.4	17.1	0.2	1.1
8	6	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
9	7	53491	53065	99.2%	11.0	-9.4	33.4	2.8	7.9
10	8	53491	53418	99.9%	10.7	-6.9	30.5	1.4	3.7
11	9	53491	53401	99.8%	10.3	-5.0	28.2	0.8	2.2
12	9	53491	53402	99.8%	9.9	-3.4	26.4	0.4	1.5
13	10	53491	53394	99.8%	9.4	-0.8	24.1	0.3	0.8
14	11	53491	53402	99.8%	9.1	0.4	22.4	0.3	0.8
15	12	53491	53406	99.8%	9.1	1.3	21.0	0.3	0.8
16	13	53491	53415	99.9%	8.6	3.6	17.9	0.3	0.8
17	6	47691	47596	99.8%	8.4	-11.3	36.3	2.0	6.2
18	7	47988	47423	98.8%	8.6	-12.4	37.3	2.3	7.2
19	8	47695	47580	99.8%	7.9	-9.3	33.9	1.3	3.7
20	9	47691	47578	99.8%	7.6	-7.1	31.1	0.7	2.0
21	10	47934	47609	99.3%	7.5	-5.0	28.9	0.4	10.3
22	10	47691	47585	99.8%	7.7	-1.7	25.9	0.3	0.8
23	11	47691	47577	99.8%	7.9	-0.0	23.6	0.3	0.7
24	12	47866	47597	99.4%	7.9	1.2	21.8	0.3	4.6
25	13	47691	47584	99.8%	7.9	3.5	18.7	0.3	0.7
26	6	53491	53147	99.4%	11.6	-13.1	38.4	2.6	7.3
27	6	53491	53332	99.7%	11.5	-12.1	37.3	2.2	6.2
28	8	53491	53413	99.9%	11.2	-9.5	33.9	1.1	3.1
29	9	53491	53395	99.8%	10.9	-7.4	31.3	0.6	1.9
30	9	53491	53389	99.8%	10.5	-5.4	29.0	0.4	1.3

Table 8.56: Cell #33 Summary Statistics (HMA, LVR, 305023) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
31	10	53491	53387	99.8%	10.2	-2.2	26.0	0.3	0.8
32	11	53491	53394	99.8%	9.7	-0.1	24.0	0.3	0.8
33	12	53491	53395	99.8%	9.7	1.0	22.2	0.3	0.7
34	13	53491	53403	99.8%	9.1	3.4	19.2	0.2	0.8
35	1	251519	250674	99.7%	13.2	-27.2	59.2	18.1	38.0
36	3	251519	251106	99.8%	14.1	-21.3	48.6	8.5	22.8
37	4	251497	247502	98.4%	14.4	-17.4	44.1	5.4	15.0
38	6	251491	249774	99.3%	14.3	-13.3	39.6	3.1	8.6
39	7	251509	250562	99.6%	14.2	-10.0	36.9	1.7	5.3
40	8	251514	250985	99.8%	13.9	-7.6	35.1	1.0	3.2

Table 8.57: Cell #34 Summary Statistics (HMA, LVR, 273523).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	6	763	383	50.2%	-0.3	-0.8	1.5	2.9	5.4
2	7	763	669	87.7%	-0.6	-1.4	2.8	0.5	2.2
3	8	865	664	76.8%	-0.9	-1.5	-0.4	0.3	3.7
4	9	763	669	87.7%	-0.9	-1.3	-0.7	0.2	0.3
5	10	763	669	87.7%	0.1	-0.1	0.3	0.2	0.2
6	10	763	669	87.7%	1.3	1.2	1.4	0.2	0.2
7	11	763	669	87.7%	4.1	3.9	4.3	0.2	0.3
8	12	774	669	86.4%	-0.4	-0.5	-0.2	0.2	0.3
9	13	29074	669	2.3%	0.0	-0.1	0.1	55.2	99.5
10	1	239003	235125	98.4%	13.9	-25.3	56.5	16.1	44.6
11	3	227541	222085	97.6%	14.5	-22.3	47.4	9.4	23.7
12	4	72997	68351	93.6%	14.4	-15.0	44.4	5.7	17.0
13	6	73000	72186	98.9%	12.9	-12.2	40.8	3.8	10.7
14	7	72997	72687	99.6%	12.5	-8.4	37.3	2.0	5.7
15	8	116982	105939	90.6%	15.9	-7.3	36.1	1.8	9.1
16	9	72995	72895	99.9%	12.4	-5.8	34.8	0.9	2.9
17	10	72989	72898	99.9%	12.5	-5.0	33.7	0.6	2.1
18	10	101065	99785	98.7%	14.1	-3.9	31.9	0.4	9.8
19	10	123342	119307	96.7%	12.9	-2.7	30.5	0.3	8.5
20	11	244316	243883	99.8%	13.3	-2.0	29.4	0.3	1.1
21	11	244322	243833	99.8%	13.2	-1.1	28.2	0.3	0.8
22	12	91797	89441	97.4%	13.8	-0.4	26.7	0.2	3.2
23	12	244354	243470	99.6%	12.8	0.2	26.1	0.2	0.6
24	13	244354	243760	99.8%	12.6	0.6	25.3	0.2	0.5
25	13	73003	72757	99.7%	12.4	1.2	23.8	0.2	0.5
26	13	89774	89162	99.3%	12.5	1.7	22.9	0.2	0.6

Table 8.58: Cell #35 Summary Statistics (HMA, LVR, 314935).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	6	59760	58164	97.3%	8.3	-14.5	37.9	2.7	17.1
2	7	59762	58731	98.3%	7.8	-13.2	36.3	2.2	11.9
3	8	59762	56471	94.5%	8.4	-10.5	33.0	1.4	13.9
4	9	59762	58373	97.7%	7.0	-8.2	30.2	0.7	11.3
5	10	59762	53252	89.1%	10.0	-6.0	28.2	0.4	9.4
6	10	59762	53252	89.1%	9.8	-2.5	25.2	0.3	9.9
7	11	59762	53252	89.1%	9.2	-0.4	23.0	0.2	8.9
8	12	59762	53252	89.1%	9.1	0.7	21.4	0.2	6.8
9	13	59762	53250	89.1%	8.6	2.8	18.9	0.2	6.3
10	6	59762	58313	97.6%	8.2	-14.9	37.5	3.2	11.1
11	7	59762	58424	97.8%	8.0	-14.0	36.4	2.8	11.2
12	8	59762	56472	94.5%	8.5	-10.7	33.1	1.5	8.4
13	9	59762	58707	98.2%	6.9	-7.8	30.3	0.7	9.2
14	9	59762	58809	98.4%	6.8	-5.9	27.9	0.4	7.7
15	10	59762	53252	89.1%	9.4	-2.6	24.7	0.3	5.9
16	11	59762	53252	89.1%	9.0	-0.4	22.7	0.2	6.2
17	12	59757	53252	89.1%	9.1	1.0	21.2	0.2	29.6
18	13	59757	53251	89.1%	8.6	3.0	18.7	0.2	27.7
19	7	6495	0	0.0%	NaN	NaN	NaN	19.0	60.9
20	8	59757	53160	89.0%	10.3	-11.0	33.3	2.6	40.7
21	9	59757	53252	89.1%	10.2	-8.1	30.5	1.4	34.9
22	9	59757	53252	89.1%	10.0	-5.6	28.4	0.6	34.3
23	10	59757	53253	89.1%	9.2	-4.1	26.1	0.4	32.9
24	11	59757	53253	89.1%	9.0	-1.3	23.7	0.3	31.1
25	12	59757	53252	89.1%	8.9	0.4	21.9	0.2	30.6
26	13	59757	53252	89.1%	8.9	1.4	20.5	0.2	29.3
27	13	59757	53251	89.1%	8.4	3.3	18.1	0.2	27.9
28	6	59757	52912	88.5%	9.8	-6.5	30.4	2.1	38.5
29	7	59757	51401	86.0%	10.4	-3.8	27.7	1.0	33.9
30	8	59757	51870	86.8%	9.7	-1.9	25.8	0.5	32.2

Table 8.59: Cell #35 Summary Statistics (HMA, LVR, 314935) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
31	10	59757	53251	89.1%	8.2	-0.1	22.4	0.3	30.3
32	10	59757	52974	88.6%	8.3	0.8	20.7	0.2	30.0
33	11	53262	53243	100.0%	8.2	1.6	19.3	0.2	0.5
34	13	53262	53242	100.0%	8.0	3.4	17.0	0.2	0.4
35	6	53262	52281	98.2%	11.2	-15.9	38.2	3.2	10.2
36	7	53262	52949	99.4%	10.7	-13.8	35.7	2.3	6.8
37	8	58898	31946	54.2%	22.6	-0.5	34.1	2.7	18.1
38	9	53262	53236	100.0%	10.4	-8.0	30.1	0.6	2.3
39	10	53262	53233	99.9%	9.8	-6.1	27.6	0.4	1.4
40	10	53262	53240	100.0%	9.4	-2.7	24.5	0.3	0.9
41	11	53262	53242	100.0%	9.1	-0.4	22.6	0.2	1.1
42	12	53262	53248	100.0%	9.1	0.9	21.0	0.2	0.7
43	13	53262	53250	100.0%	8.6	3.1	18.5	0.2	0.4
44	6	53262	52281	98.2%	11.2	-15.6	37.8	3.1	9.8
45	7	53262	52941	99.4%	10.8	-13.7	35.6	2.3	6.7
46	8	53262	53234	99.9%	10.4	-10.5	32.4	1.1	3.8
47	9	53262	53223	99.9%	10.0	-8.1	29.7	0.6	3.4
48	9	53262	53225	99.9%	9.7	-6.1	27.6	0.4	2.2
49	10	50947	33375	65.5%	3.9	-2.6	26.5	0.6	22.2
50	11	50980	31474	61.7%	3.6	-0.3	20.9	0.4	20.9
51	12	51089	31379	61.4%	4.2	0.9	19.1	0.4	18.0
52	13	51084	32903	64.4%	5.6	2.9	16.0	0.5	14.5
53	6	50845	42157	82.9%	10.3	-15.5	43.3	4.2	34.5
54	7	50843	42118	82.8%	10.1	-14.1	39.9	3.5	33.9
55	8	50892	40468	79.5%	7.3	-11.1	34.9	1.9	29.7
56	9	50945	41091	80.7%	6.3	-8.3	32.5	1.1	25.6
57	9	50933	40583	79.7%	5.6	-6.3	30.9	0.8	24.2
58	10	51003	33418	65.5%	3.9	-2.9	23.1	0.6	21.8
59	11	50868	30937	60.8%	3.5	-0.7	21.3	0.5	18.3
60	12	51098	31093	60.8%	4.1	0.7	19.5	0.4	17.3

Table 8.60: Cell #35 Summary Statistics (HMA, LVR, 314935) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
61	13	51143	30810	60.2%	5.5	3.2	14.9	0.4	14.5
62	1	253162	252067	99.6%	12.4	-26.2	57.4	16.8	36.8
63	3	253159	251782	99.5%	13.2	-22.8	50.8	10.5	26.0
64	4	253120	247986	98.0%	13.6	-18.9	45.7	6.3	18.2
65	6	253108	250770	99.1%	13.5	-13.9	40.4	3.4	9.5
66	7	253162	251886	99.5%	13.4	-10.3	37.0	1.8	8.0
67	8	253156	251915	99.5%	13.2	-9.3	36.1	1.4	5.2
68	9	253155	252127	99.6%	13.1	-7.7	34.6	0.8	3.5

Table 8.61: Cell #36 Summary Statistics (PCC, LVR, 400046).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	400046	396702	99.2%	11.9	-29.5	49.9	12.2	27.3
2	3	400046	398522	99.6%	12.0	-27.9	46.9	9.5	22.6
3	5	400044	394842	98.7%	12.5	-25.1	41.5	5.7	17.1
4	6	332600	326932	98.3%	13.0	-19.0	38.9	3.8	10.0
5	7	332600	327057	98.3%	13.1	-17.1	37.4	2.8	8.2
6	8	332600	326876	98.3%	13.3	-14.0	35.4	1.7	5.2
7	9	332600	326067	98.0%	13.1	-12.1	33.4	1.0	3.1
8	10	332600	329455	99.1%	12.5	-6.3	30.2	0.5	1.6
9	11	108223	106293	98.2%	12.1	-2.5	26.1	0.4	1.7
10	12	332600	331563	99.7%	12.0	-0.8	26.0	0.4	1.2
11	13	122602	121281	98.9%	11.1	2.0	21.0	0.4	1.2

Table 8.62: Cell #37 Summary Statistics (PCC, LVR, 400271).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	5	400271	396194	99.0%	12.4	-23.2	41.5	5.2	13.7
2	3	400269	398500	99.6%	12.1	-26.6	46.5	9.1	21.9
3	1	400271	397011	99.2%	12.0	-28.8	50.0	12.4	27.4
4	7	332306	328336	98.8%	12.8	-15.1	38.7	3.3	9.1
5	7	332306	328390	98.8%	13.0	-13.4	37.3	2.5	7.4
6	8	332306	330669	99.5%	12.9	-10.8	35.4	1.5	5.0
7	9	332306	330079	99.3%	12.6	-8.1	33.3	0.8	5.3
8	10	332306	330952	99.6%	12.2	-4.0	30.3	0.5	2.8
9	11	332306	330067	99.3%	12.2	-1.8	28.2	0.4	1.8
10	12	332306	330145	99.3%	12.1	-0.6	26.3	0.4	3.1
11	13	332306	331186	99.7%	11.5	1.6	22.2	0.4	0.9

Table 8.63: Cell #38 Summary Statistics (PCC, LVR, 399627).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	399627	362977	90.8%	11.6	-28.9	50.4	12.0	28.0
2	3	399626	368652	92.2%	11.3	-26.5	46.2	8.3	21.2
3	5	399627	372080	93.1%	11.8	-23.5	42.2	5.1	14.6
4	6	331535	306073	92.3%	12.1	-15.4	39.5	3.6	10.7
5	7	331535	294341	88.8%	12.9	-13.3	37.7	2.6	11.4
6	8	331535	298877	90.1%	12.4	-16.8	35.1	1.5	24.6
7	9	331535	294131	88.7%	12.5	-13.7	32.7	0.8	21.4
8	10	331535	290729	87.7%	12.2	-5.1	29.7	0.5	13.8
9	11	331131	292353	88.3%	12.0	-2.9	27.2	0.5	8.5
10	12	331131	295203	89.1%	11.5	-3.6	25.3	0.4	6.6
11	13	331535	296874	89.5%	11.3	-0.9	20.9	0.4	2.3

Table 8.64: Cell #39 Summary Statistics (PCC, LVR, 399939).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	399936	397534	99.4%	11.0	-29.4	47.9	11.7	24.6
2	3	230326	228605	99.3%	11.6	-27.2	43.0	8.4	19.2
3	5	399936	396414	99.1%	11.6	-23.8	40.4	4.9	12.8
4	7	331933	326612	98.4%	12.2	-16.1	37.8	3.3	9.0
5	7	331933	327729	98.7%	12.3	-13.8	36.5	2.3	7.1
6	8	331933	328288	98.9%	12.3	-10.6	34.1	1.1	4.2
7	9	331933	328314	98.9%	12.1	-8.1	32.0	0.6	2.6
8	10	331933	329078	99.1%	11.5	-4.5	28.9	0.3	1.7
9	11	331933	329885	99.4%	11.6	-2.4	26.9	0.4	3.4
10	12	127070	123449	97.2%	12.1	-1.1	24.0	0.3	0.9
11	13	331933	330549	99.6%	11.0	1.8	20.7	0.3	1.0
12	1	399936	397277	99.3%	11.4	-29.1	49.7	12.7	26.9
13	3	399936	398653	99.7%	11.5	-27.0	45.7	8.8	20.9
14	5	399936	396343	99.1%	11.8	-23.5	40.9	4.9	12.9
15	6	331933	327849	98.8%	12.3	-16.8	39.4	3.8	10.4
16	7	331933	327564	98.7%	12.3	-14.9	37.5	2.8	7.9
17	8	331936	328330	98.9%	12.2	-11.4	34.7	1.3	4.7
18	9	331936	328136	98.9%	12.2	-8.8	32.6	0.7	2.7
19	10	331936	328825	99.1%	11.7	-5.1	29.7	0.3	2.6
20	11	331936	330214	99.5%	11.6	-3.0	27.2	0.3	2.6
21	12	331936	330329	99.5%	11.5	-1.2	25.4	0.3	3.9
22	13	331936	330580	99.6%	11.2	2.0	21.0	0.2	0.9
23	1	399928	392864	98.2%	11.2	-27.7	59.8	15.5	42.4
24	3	399936	396565	99.2%	12.0	-24.6	52.1	10.4	29.6
25	5	399939	396192	99.1%	12.7	-22.6	43.8	5.6	16.2
26	6	399939	388290	97.1%	12.3	-21.0	46.5	6.0	21.5
27	7	331936	320403	96.5%	12.6	-17.3	42.9	4.6	14.6
28	8	331936	328671	99.0%	13.0	-13.3	38.7	2.4	8.8
29	9	331936	328533	99.0%	12.9	-10.2	35.9	1.2	4.6
30	10	331936	328906	99.1%	12.4	-5.6	31.7	0.4	2.0

Table 8.65: Cell #39 Summary Statistics (PCC, LVR, 399939) continued

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
31	11	331936	330521	99.6%	11.9	-3.1	28.9	0.3	2.8
32	12	331936	330529	99.6%	11.6	-1.3	26.4	0.3	1.8
33	13	331936	330402	99.5%	10.9	1.7	21.1	0.2	2.8
34	3	399939	365224	91.3%	10.1	-17.4	45.6	8.1	32.1
35	5	298085	261369	87.7%	6.0	-17.8	43.8	5.4	35.7
36	6	252769	247546	97.9%	9.0	-12.9	35.8	2.7	24.3
37	8	252769	247546	97.9%	9.5	-9.9	30.3	1.1	6.6
38	9	252769	250317	99.0%	9.5	-7.5	28.9	0.6	3.6
39	10	252769	250363	99.0%	9.3	-3.5	26.6	0.3	2.1
40	11	252769	251634	99.6%	9.3	-2.3	24.6	0.4	1.7
41	12	252769	251488	99.5%	9.3	-0.8	22.6	0.3	0.9
42	13	230365	226315	98.2%	9.6	1.5	18.4	0.2	1.8

Table 8.66: Cell #40 Summary Statistics (PCC, LVR, 395374).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	395374	392039	99.2%	11.7	-28.7	49.8	12.6	27.2
2	3	395372	392901	99.4%	11.8	-26.6	46.1	9.1	21.3
3	5	395368	390658	98.8%	12.1	-23.5	41.7	5.5	14.3
4	6	395368	389732	98.6%	12.1	-20.8	38.3	3.5	9.4
5	7	331551	326819	98.6%	12.5	-14.4	36.8	2.5	7.1
6	8	331551	328441	99.1%	12.4	-10.8	34.5	1.3	4.3
7	9	331551	328762	99.2%	12.1	-8.6	32.2	0.7	2.6
8	10	331551	328790	99.2%	11.7	-4.6	29.2	0.4	1.5
9	11	331551	329532	99.4%	11.7	-2.3	26.8	0.3	1.1
10	12	331551	330096	99.6%	11.3	-0.9	24.9	0.3	0.9
11	13	331551	330152	99.6%	11.0	1.7	21.2	0.2	0.9

Table 8.67: Cell #52 Summary Statistics (PCC, LVR, 247195).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	247195	244717	99.0%	10.3	-23.5	44.8	9.7	22.7
2	2	247195	246062	99.5%	10.6	-23.5	44.8	9.7	22.7
3	2	247195	245954	99.5%	10.6	-22.5	43.4	8.5	20.4
4	3	247195	246002	99.5%	10.5	-21.7	42.3	7.7	18.8
5	4	247195	246077	99.5%	10.5	-19.6	40.6	6.0	15.5
6	6	247195	244982	99.1%	10.9	-17.1	37.8	3.9	10.9
7	7	247195	241063	97.5%	11.1	-14.7	35.2	2.4	7.5
8	8	247195	241234	97.6%	11.2	-12.2	33.7	1.6	5.8
9	1	247195	245280	99.2%	11.2	-24.3	47.7	11.3	23.7
10	2	247195	245948	99.5%	12.6	-23.2	47.8	10.5	22.8
11	2	247195	245910	99.5%	12.5	-22.6	46.5	9.6	20.7
12	3	247195	246069	99.5%	11.7	-21.5	43.5	8.1	18.1
13	4	247195	246021	99.5%	11.3	-20.2	41.8	6.7	15.4
14	6	247195	244994	99.1%	11.4	-18.5	39.7	5.0	11.9
15	7	247195	240535	97.3%	11.8	-15.2	36.3	2.8	7.9
16	8	247195	243051	98.3%	11.5	-12.8	34.3	1.9	5.7
17	1	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
18	1	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
19	1	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
20	2	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
21	2	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
22	3	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
23	5	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
24	5	0	0	0.0%	NaN	NaN	NaN	NaN	NaN

Table 8.68: Cell #53 Summary Statistics (PCC, LVR, 245835).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	245835	243643	99.1%	11.3	-25.2	47.3	11.8	25.1
2	1	245835	243175	98.9%	11.6	-24.6	45.8	10.9	23.5
3	2	245835	244244	99.4%	11.4	-23.8	44.4	9.9	21.8
4	3	245835	243980	99.2%	11.1	-22.9	42.5	8.6	19.4
5	4	245835	243802	99.2%	11.4	-21.2	40.8	6.8	15.7
6	5	245835	243740	99.1%	11.5	-18.6	38.6	4.9	11.8
7	7	245835	240941	98.0%	11.5	-14.8	35.3	2.6	8.2
8	8	245835	243189	98.9%	11.3	-12.7	33.7	1.8	5.7
9	1	245835	243526	99.1%	11.6	-24.9	47.2	11.0	23.0
10	2	245835	244156	99.3%	11.7	-24.1	46.0	9.9	21.3
11	2	245835	243920	99.2%	12.5	-23.5	46.7	9.3	20.3
12	3	245835	244160	99.3%	11.6	-22.5	43.6	8.0	17.9
13	4	245835	244164	99.3%	11.5	-21.2	41.9	6.5	15.0
14	6	245835	243143	98.9%	11.7	-18.6	39.2	4.5	11.1
15	7	245835	240703	97.9%	11.9	-15.0	35.9	2.6	7.5
16	8	245835	241578	98.3%	11.8	-12.5	34.2	1.7	5.5

Table 8.69: Cell #54 Summary Statistics (PCC, LVR, 97126).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	97122	96892	99.8%	12.2	-24.7	45.7	12.2	23.5
2	2	97123	96875	99.7%	12.3	-24.5	44.0	10.4	20.0
3	3	97123	96914	99.8%	12.4	-22.9	42.0	8.7	16.9
4	4	97123	96815	99.7%	12.4	-23.2	40.6	7.9	15.0
5	5	97121	97007	99.9%	12.5	-22.6	39.6	7.1	14.8
6	6	97116	96789	99.7%	13.0	-18.8	37.7	4.3	9.7
7	7	97101	96211	99.1%	12.9	-16.1	35.2	2.9	7.3
8	1	97121	96771	99.6%	12.1	-25.9	43.7	11.4	21.0
9	2	97123	96787	99.7%	12.3	-24.8	42.3	9.8	18.7
10	3	97120	96859	99.7%	12.6	-24.4	40.9	8.5	17.8
11	4	97123	96719	99.6%	12.4	-22.7	39.7	7.3	15.3
12	5	97120	97003	99.9%	12.5	-22.1	39.1	6.8	13.0
13	6	97116	96844	99.7%	12.9	-19.2	37.3	4.3	9.8
14	7	97114	96511	99.4%	13.0	-15.7	34.5	2.6	6.3
15	8	97116	97021	99.9%	12.1	-10.7	31.0	1.3	4.4
16	8	97110	97012	99.9%	12.2	-11.7	31.7	1.4	3.9

Table 8.70: Cell #60 Summary Statistics (HMA, ML, 102654).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	102653	102073	99.4%	11.8	-23.8	52.5	12.8	29.1
2	1	102650	102069	99.4%	11.8	-23.3	51.7	12.0	27.1
3	2	102651	102469	99.8%	12.0	-21.8	49.2	9.7	22.5
4	3	102653	102179	99.5%	12.2	-20.1	46.8	7.7	18.1
5	4	102652	102441	99.8%	12.2	-19.7	46.2	7.4	17.4
6	6	102646	102550	99.9%	12.3	-15.7	41.5	4.1	9.6
7	7	102646	102357	99.7%	12.2	-13.0	38.3	2.5	6.7
8	1	102653	102369	99.7%	12.4	-23.5	55.1	15.1	31.6
9	1	102653	102266	99.6%	12.3	-22.9	53.6	13.6	29.0
10	2	102651	102462	99.8%	12.5	-21.2	50.1	10.5	23.5
11	3	102648	102373	99.7%	12.6	-19.7	47.6	8.5	19.7
12	3	102649	102373	99.7%	12.6	-19.3	47.0	7.9	18.8
13	6	102651	102558	99.9%	12.5	-15.9	42.1	4.8	11.9
14	7	102648	102444	99.8%	12.4	-12.8	38.3	2.6	6.6

Table 8.71: Cell #62 Summary Statistics (HMA, ML, 93694).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	93692	93029	99.3%	13.5	-23.6	52.2	13.5	28.4
2	1	93689	92931	99.2%	13.5	-23.1	51.0	12.4	26.1
3	2	93690	93409	99.7%	13.6	-21.8	49.0	10.5	22.9
4	3	93689	93208	99.5%	13.8	-21.1	47.5	9.2	20.3
5	3	93691	93210	99.5%	13.9	-20.4	47.0	8.7	19.4
6	6	93684	93470	99.8%	14.1	-16.0	41.3	4.4	14.3
7	7	93690	93155	99.4%	13.8	-13.3	37.6	2.4	11.3
8	1	93689	93316	99.6%	14.2	-23.6	54.2	15.6	31.1
9	1	93691	93224	99.5%	14.5	-22.8	52.7	13.9	28.6
10	2	93690	93416	99.7%	14.4	-21.8	50.7	12.1	25.3
11	3	93689	93314	99.6%	14.4	-20.7	48.9	10.5	22.5
12	3	93691	93314	99.6%	14.5	-20.4	48.2	10.0	21.5
13	6	93685	93477	99.8%	14.5	-15.9	42.1	5.3	14.9
14	7	93672	93240	99.5%	14.2	-13.2	38.0	2.7	11.7

Table 8.72: Cell #93 Summary Statistics (HMA, ML, 225905).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	46783	30080	64.3%	15.7	-21.2	45.5	15.5	262.8
2	1	221691	219286	98.9%	11.8	-26.8	51.4	12.6	28.4
3	2	224712	221902	98.7%	11.3	-25.7	49.2	10.6	26.0
4	3	224889	223540	99.4%	11.6	-23.6	47.3	9.1	22.4
5	3	224802	222365	98.9%	11.7	-23.6	46.5	7.7	21.4
6	6	191088	189496	99.2%	11.7	-18.1	38.6	4.5	19.4
7	7	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
8	1	0	0	0.0%	NaN	NaN	NaN	NaN	NaN
9	1	225809	224362	99.4%	11.2	-25.8	47.9	10.7	25.4
10	2	225144	223533	99.3%	10.9	-25.4	46.4	9.7	23.8
11	3	225094	223562	99.3%	11.2	-24.1	44.5	8.2	30.1
12	3	225639	224469	99.5%	11.2	-24.0	43.9	7.9	20.7
13	6	224716	222779	99.1%	11.6	-18.4	37.7	4.2	12.3
14	7	225813	224464	99.4%	11.9	-13.5	33.1	2.2	8.6

Table 8.73: Cell #95 Summary Statistics (HMA, ML, 225263).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	184505	161528	87.5%	11.1	-22.6	52.6	13.9	52.3
2	1	221672	219213	98.9%	11.8	-25.2	50.5	12.1	28.5
3	2	128947	125997	97.7%	12.0	-20.6	45.2	10.5	24.6
4	3	225261	224767	99.8%	11.9	-23.4	47.5	10.1	24.2
5	6	225263	224315	99.6%	12.3	-17.0	38.0	4.3	11.2
6	7	141078	137555	97.5%	12.2	-11.3	32.6	2.0	939.2
7	1	225261	224479	99.7%	10.8	-26.7	48.6	12.6	29.4
8	2	225092	224037	99.5%	11.2	-25.6	47.1	10.9	25.9
9	3	225089	224014	99.5%	11.2	-25.0	46.3	10.1	24.7
10	3	225090	224021	99.5%	11.4	-24.2	45.4	9.3	22.7
11	6	225114	223699	99.4%	11.9	-18.4	38.6	4.7	12.5
12	7	225260	224452	99.6%	12.1	-13.1	33.4	2.2	9.5

Table 8.74: Cell #97 Summary Statistics (HMA, ML, 334648).

sensor	group	number active	number unflagged	% good	median [°C]	min [°C]	max [°C]	median daily range [°C]	max daily range [°C]
1	1	208288	189142	90.8%	12.7	-21.4	51.3	10.4	187.2
2	2	334552	332598	99.4%	13.3	-21.8	50.5	9.7	25.0
3	3	181039	172150	95.1%	12.7	-18.5	46.7	9.1	39.1
4	4	334619	326816	97.7%	12.9	-25.1	52.8	10.2	28.8
5	5	311955	306923	98.4%	12.9	-19.4	47.1	7.2	110.3
6	5	166872	161936	97.0%	14.1	-16.5	43.5	6.9	30.9
7	7	269130	234137	87.0%	11.3	-22.2	39.7	6.1	40.1
8	7	325904	318899	97.9%	13.0	-13.2	38.6	4.3	23.0
9	1	334547	320742	95.9%	11.7	-21.6	50.1	9.0	34.1
10	1	334563	301868	90.2%	11.9	-20.4	48.4	6.5	33.2
11	3	334541	312939	93.5%	13.1	-19.1	45.7	4.3	23.7
12	3	186721	166644	89.2%	14.9	-17.9	49.6	8.9	36.4
13	4	233697	210873	90.2%	13.4	-17.2	43.4	7.5	40.1
14	5	334561	333422	99.7%	12.2	-25.3	50.5	9.1	28.4
15	6	211130	154490	73.2%	14.1	-13.9	45.7	6.2	61.7
16	7	334536	331675	99.1%	12.5	-20.7	45.4	5.4	24.0

References

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover Publications, Inc., New York, 1965. 1046 pp.

[2] J. R. Benjamin and C. A. Cornell. *Probability, Statistics, and Decision for Civil Engineers*. McGraw-Hill Book Company, New York, 1970. 684 pp.

[3] P. Bloomfield. *Fourier Analysis of Time Series: An Introduction*. John Wiley & Sons, Inc., New York, 1976. ISBN 0-471-08256-2. 258 pp.

[4] S. Dai and D. Van Deusen. *Digital signal processing for MnROAD offline data.* Technical Report MN-RC-1996-09, Minnesota Department of Transportation, Saint Paul, MN, 1996.

[5] H. A. David. *Order Statistics*. John Wiley and Sons, Inc., New York, second edition, 1981. ISBN 0-471-902723-5. 360 pp.

[6] D. Van Deusen. *Selection of flexible backcalculation software for the minnesota road research project.* Technical Report MN-RC-1996-29, Minnesota Department of Transportation, Saint Paul, MN, 1996.

[7] J. J. Devore. *Proability and Statistics for Engineering and the Sciences*. Duxbury, Pacific Grove, CA, 2000. 775 pp.

[8] S. Dogru, V. Rajan, K. Rieck, J. R. Slagle, B. S. Tjan, and Y. Wang. *Dfql: A graphical data flow query language for retrieval, analysis, and visualization of a scientific database.* Technical Report MN-RC-1996-02, Minnesota Department of Transportation, Saint Paul, MN, 1996.

[9] P. J. Huber. *Robust Statistics*. John Wiley and Sons, Inc., New York, 1981. ISBN 0-471-41805-6. 308 pp.

[10] N. L. Johnson and S. Kotz. *Continuous Univariate Distributions - 1*. John Wiley and Sons, Inc., New York, 1970. ISBN 0-471-44626-2. 300 pp.

[11] A. Koubaa and H. Stolarski. *Assistance in the validation of the MnROAD database.* Technical Report unpublished Report to the Office of Materials and Road Research, Minnesota Department of Transportation, Saint Paul, MN, 2003.

[12] W. C. Lau and M. S. Alouini. *Development of automated procedures for dynamic megadac calibration and data recovery.* Technical Report unpublished Report to the Office of Materials and Road Research, Minnesota Department of Transportation, Saint Paul, MN, 2002.

[13] The MathWorks. Matlab datenum, 2009. URL http://www.mathworks.com/access/helpdesk/help/techdoc/ref/datenum.html. Online; accessed 01-September-2009.

[14] The MathWorks. Matlab file formats, 2009. URL http://www.mathworks.com/access/helpdesk/help/techdoc/ref/fileformats.html. Online; accessed 01-September-2009.

[15] The MathWorks. Matlab mat files, 2009. URL http://www.mathworks.com/access/helpdesk/help/techdoc/import_export/braidzi-1.html. Online; accessed 01-September-2009.

[16] Mn/DOT. MnROAD brochure, 2009. URL http://www.dot.state.mn.us/mnroad/pdfs/mnroad_brochure.pdf. Online; accessed 01-September-2009.

[17] Mn/DOT. Map of facility, 2009. URL http://www.dot.state.mn.us/mnroad/images/facility.jpg. Online; accessed 01-September-2009.

[18] Mn/DOT. MnROAD funding, 2009. URL http://www.dot.state.mn.us/mnroad/funding.html. Online; accessed 01-September-2009.

[19] Mn/DOT. MnROAD history, 2009. URL http://www.dot.state.mn.us/mnroad/history/. Online; accessed 01-September-2009.

[20] D. C. Montgomery and G. C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, Inc., New York, 1994. 895 pp.

[21] F. Mosteller and J. W. Tukey. *Data Analysis and Regression*. Addison Wesley, New York, 1977. ISBN 978-0201048544. 588 pp.

[22] S. Nadarajah. "A generalized normal distribution." *Journal of Applied Statistics*, 32(7):685–694, 2005.

[23] M. B. Priestley. *Spectral Analysis of Time Series*. Academic Press, New York, 1983. ISBN 0-125-64922-3. 890 pp.

[24] J. Prins, D. McCormack, D. Michelson, and K. Horrell. Nist/sematech e-handbook of statistical methods, 2009. URL http://www.itl.nist.gov/div898/handbook/prc/section2/prc252.htm. Online; accessed 01-September-2009.

[25] B. Stroustrup. *Programming: Principles and Practice Using C++*. Addison-Wesley, New York, 2008. ISBN 978-0321543721. 1272 pp.

[26] J. W. Tukey. *Exploratory Data Analysis*. Addison Wesley, New York, 1977. ISBN 978-0201076165. 688 pp.

[27] R. Walpole, R. Myers, S. Myers, and K. Ye. *Probability and Statistics for Engineers and Scientists*. Prentice Hall, Upper Saddle River, NJ, seventh edition, 2002. 730 pp.

[28] Wikipedia. Average — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/Average. Online; accessed 01-September-2009.

[29] Wikipedia. Expected value — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/Expected_value. Online; accessed 01-September-2009.

[30] Wikipedia. Generalized normal distribution — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/Generalized_normal_distribution. Online; accessed 01-September-2009.

[31] Wikipedia. Interquartile range — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/Inter-quartile_range. Online; accessed 01-September-2009.

[32] Wikipedia. Kurtosis — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/Kurtosis. Online; accessed 01-September-2009.

[33] Wikipedia. Laplace distribution — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/Laplace_distribution. Online; accessed 01-September-2009.

[34] Wikipedia. Lognormal distribution — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/Log-normal_distribution. Online; accessed 01-September-2009.

[35] Wikipedia. Mean — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/Mean. Online; accessed 01-September-2009.

[36] Wikipedia. Median — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/Median. Online; accessed 01-September-2009.

[37] Wikipedia. Normal distribution — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/Normal_distribution. Online; accessed 01-September-2009.

[38] Wikipedia. Percentile — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/Percentile. Online; accessed 01-September-2009.

[39] Wikipedia. Quantile — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/Quantile. Online; accessed 01-September-2009.

[40] Wikipedia. Skewness — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/Skewness. Online; accessed 01-September-2009.

[41] Wikipedia. Standard deviation — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/Standard_deviation. Online; accessed 01-September-2009.

[42] Wikipedia. Variance — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/Variance. Online; accessed 01-September-2009.

[43] Wikipedia. ASCII — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/ASCII. Online; accessed 01-September-2009.

[44] Wikipedia. Comma-separated values — Wikipedia, the free encyclopedia, 2009. URL http://en.wikipedia.org/wiki/Comma-separated_values. Online; accessed 01-September-2009.

[45] B. Worel. *Guide to the MnROAD database. supplementary unpublished report for lessons learned project.* Technical Report Contract 81655, Minnesota Department of Transportation, Saint Paul, MN, 2006.

[46] S. Zhang and J. Jin. *Computation of Special Functions*. John Wiley and Sons, Inc., New York, 1996. 717 pp.

Appendix A

User's Guide

A.1 Overview

A.1.1 `Analysis` Folder

All of the *Matlab* code that drives and evaluates the MnROAD data is stored within the `Analysis` folder. There are 24 representative *Matlab* functions, which are denoted as such by the '`.m`' file extension, inside the `Analysis` folder:

- `CellProperties.m`
- `CreateDailyRanges.m`
- `CreateInitialization.m`
- `CreateMatFiles.m`
- `CreateSubsetRanges.m`
- `CreateDailyExtremes.m`
- `Flag DailyRange.m`
- `Flag Deactivated.m`
- `Flag InconsistentDay.m`
- `Flag InconsistentMonth.m`
- `Flag InconsistentWeek.m`
- `Flag IntermittentData.m`
- `Flag LagOneOutliers.m`
- `Flag MissingData.m`
- `Flag NeighborhoodOutliers.m`
- `Flag NotYetOperational.m`
- `Flag OutOfRange.m`
- `Flag PointExtremes.m`
- `Flag TooSparseDay.m`
- `GeneralizedNormalFit.m`
- `PlotFlags.m`
- `PlotTemperatures.m`

- ResetAllFlags.m
- ResetFlags.m

The other files in the Analysis folder are the Initialization.mat file created by the CreateInitialization and an example driver function called A_RunAll.m.

A.1.2 Statistics Folder

All of the *Matlab* code used in the computation of the summary statistics and post-processing report generation is stored within the Statistics folder. There are two categories of functions in this folder: (1) operational code and (2) helper routines. The helper routines are called by the operational code, and it is anticipated that these helper routines will never be directly executed by a user.

The operational code category includes 5 *Matlab* functions, which are denoted as such by the '.m' file extension:

- CreateGradientStats.m
- CreateSensorStats.m
- DepthStatCall.m
- GradientStatCall.m
- SensorStatCall.m

The helper routines category includes

- nanmean_withflags.m
- nanmedian_withflags.m
- nanquartiles.m
- nanquartiles_withflags.m
- nanstats.m
- nanstd.m
- nansum_withflags.m
- nantrimmean.m
- nanvar_withflags.m

A.1.3 `Data` Folder

All of the MnROAD data is to be stored inside the `Data` folder. However, as is explained above, the `CreatMatFiles` function in the `Analysis` folder is to be used to create properly formatted data files (in a compressed binary `.mat` form). This is useful for many reasons, but primarily because it combines the cell data from every year to create a single file for each cell. This allows for much more streamlined and efficient code.

The `Data` folder also includes four the four binary data files `DailyRanges.mat`, `SubsetRanges.mat`, `GradientStats.mat`, and `SensorStats.mat`. The first two of these are created during the execution of the analysis code (i.e. flagging and testing), the second two are created during the post-processing.

All of the files in the `Data` folder are cannot to be modified directly using a standard text editor; they are linked to the various test and flagging functions. These files can, however, be opened and reviewed directly in MATLAB.

A.1.4 `Raw Data Oct07` Folder

At the beginning of this project, the University research team received two CD's containing 684 raw data files, which comprised approximately 4.4 Gigabytes of raw data. There are 48 Cells, 148 Sensor Trees, for 1,313 individual sensors. This folder contains all of the the original data.

A.2 Analysis Functions

A.2.1 `CellProperties.m`

```
%-----------------------------------------------------------------------
% usage: [type, loc, mat, thick] = CellProperties(cellno, depth)
%
%   This function returns the material properties for the specified cell
%   number at specified depth.
%
%   The arguments cellno and depth may be scalars, vectors, or matrices.
%   A scalar input for either of the two arguments is interpreted as a
%   constant matrix, equal to the input value, and of the same size as the
%   other input argument.  If both arguments are non-scalars then they must
%   be of equal size.
%
%   The four return values will be arrays of the common input size.
%
% Arguments:
%   cell         cell number of interest
%   depth        depth [ft] of interest
%
% Returns:
%   type         pavement type: 0=cell not found, 1=HMA, 2=PCC, 3=Other
%   loc          cell location: 1=ML, 2=LVR
%   mat          material: 3-letter abbreviation for material type at the
%                requested depth. This is a cell array, and not a
%                standard matrix.
```

```
%   thick       thickness [ft] of the layer at the requested depth
%
% Output:
%   NONE
%
% Preconditions:
%   NONE
%
% Notes:
% o This function is designed as a stand-alone helper function.
%
% o It is anticipated that this function will be replaced by a more generic
%   routine that automatically ties into the standard data base.
%
% o As new cells are added to the system, this function will have to change
%   to incorporate the new data.  We attempted to partition the mutable
%   information from the automated initialization.  The changing parts go
%   in this function, the more static parts go in CreateInitialization.m.
%
% o If the input cell number is not found, the return values are
%
%       [0, 0, ' ', 0]
%
%   The pavement type = 0 should be used to identify an out-of-bounds
%   input.
%
% o In the few cases where a cell has multiple properties listed, only
%   the first set are returned.
%
% o The material type abbreviations used are as follows:
%
%   TOP LAYER
%       AST     ASTRO TURF
%       04M     2004 MICRO
%       03M     2003 MICRO/MINIMAC
%       99M     1999 MICRO
%       SLU     SLURRY
%       2XC     DOUBLE CHIP SEAL
%       OIL     OIL GRAVEL
%       FDR     FULL DEPTH RECYCLE
%       WTO     WHITE TOPPING
%       CON     CONCRETE
%       HMA     HOT MIX ASPHALT
%       PCC     PLASTIC PCC STEEL CULVERTS
%       PSA     PSAB DRAINABLE BASE
%       CL6     CLASS-6 SP
%       CL5     CLASS-5 SP
%       CL4     CLASS-4 SP
%       CL3     CLASS-3 SP
%       C1F     CLASS-1F
%       C1C     CLASS-1C
%       RHM     RECLAIMED HMA
%       LAS     LARGE STONE
%       CLA     CLAY
```

```
%   BOTTOM LAYER
%
% o These 3-letter abbreviations can be replaced without adversely
%   influencing any subsequent calculations.
%
% o The clay layer at the bottom is assumed to be semi-infinite.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%-------------------------------------------------------------------------
```

A.2.2 CreateDailyRanges.m

```
%-------------------------------------------------------------------------
% usage: [] = CreateDailyRanges()
%
%   Compute daily temperature minima, maxima, and ranges for all sensors
%   in all cells for all days.  The results are written to a file.
%
% Arguments:
%   NONE
%
% Returns:
%   NONE
%
% Output:
% o Creates a file 'DailyRange.mat' in the source root directory.
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The binary source files (i.e. Cellxxx.mat) must exist in the source
%   folder identified in the initialization file.
%
% o The following flags have already been correctly applied:
%
%       FLAG_MISSING_DATA           missing data
%       FLAG_NOT_YET_OPERATIONAL    missing data at the beginning
%       FLAG_DEACTIVATED            missing data at the end
%       FLAG_TOO_SPARSE_DAY         not enough data in any day
%       FLAG_OUT_OF_RANGE           sensor outliers with annual & diurnal fit
%
% Notes:
% o The defining component of 'DailyRanges.mat' is a large matrix [R].
%
%       R(:,1)  date
%       R(:,2)  count
%       R(:,3)  minimum temperature for the day
%       R(:,4)  maximum temperature for the day
%       R(:,5)  temperature range
%       R(:,6)  cell number
```

```
%       R(:,7)  sensor
%       R(:,8)  cell type
%       R(:,9)  sensor group
%       R(:,10) cell location
%       R(:,11) month
%
% o To assist in keeping the columns straight, even if the contents changes
%   over time, the following manifest constants are defined in this routine
%   and sorted in the DailyRange.mat file.
%
%       R_DATE      = 1;    % integer day
%       R_COUNT     = 2;    % # of data in this range
%       R_MIN       = 3;    % daily minimum
%       R_MAX       = 4;    % daily maximum
%       R_RANGE     = 5;    % daily range
%       R_CELLNO    = 6;    % cell number
%       R_SENSOR    = 7;    % sensor number
%       R_TYPE      = 8;    % cell’s pavement type [1=HMA, 2=PCC, 3=Other]
%       R_GROUP     = 9;    % sensor’s depth group [1 ... 13]
%       R_LOC       = 10;   % cell’s location [1=ML, 2=LVR]
%       R_MONTH     = 11;   % month [1 ... 12]
%
% o These statistics are computed ignoring all data with one of the
%   following flags set:
%
%       FLAG_MISSING_DATA           missing data
%       FLAG_NOT_YET_OPERATIONAL    missing data at the beginning
%       FLAG_DEACTIVATED            missing data at the end
%       FLAG_TOO_SPARSE_DAY         not enough data in any day
%       FLAG_OUT_OF_RANGE           sensor outliers with annual & diurnal fit
%
%   This set of flags MUST be the same in all on the following three
%   functions:
%
%       CreateDailyRange.m
%       Flag_DailyExtremes.m
%       Flag_DailyRange.m.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%--------------------------------------------------------------------------
```

A.2.3 CreateInitialization.m

```
%--------------------------------------------------------------------------
% usage: [] = CreateInitialization( rawroot, srcroot )
%
%   This function creates the ’Initialization.mat’ file.  This binary file
%   contains the entire suite of initialization, localization, scaling, and
%   calibration constants for the entire system.
%
% Arguments:
```

```
%   rawroot     The path to the raw root data folder enclosed in single
%               quotes.  This path can be absolute or relative to the
%               current working directory.  For example,
%
%                   '../Raw Data Oct 07'.
%
%               This path should NOT end with a '/' character.
%
%               The rawroot folder contains the raw, ASCII, source data
%               files. The embraced data structure was defined by MNDOT,
%               and is defined more fully in the preconditions below.
%
%   srcroot     The path to the raw root data folder enclosed in single
%               quotes.  This path can be absolute or relative to the
%               current working directory.  For example,
%
%                   '../Data'.
%
%               This path should NOT end with a '/' character.
%
%               The srcroot folder will contain the binary source data
%               files in a MATLAB specific format (.mat), and various other
%               generated binary files.  This is the main repository for
%               this system.
%
% Returns:
%   NONE
%
% Output:
%   The values are exported in the binary file "Initialization.mat", which
%   is placed in the current working directory.
%
% Preconditions:
%
% o The raw data files are in-place as a folder hierarchy off of the
%   RAW_ROOT folder. The raw data structure is organized like:
%
%   RAW_ROOT
%       Cell1
%           1996Cell1.csv
%           1997Cell1.csv
%           ...
%           2007Cell1.csv
%           Cell1SensorsGroups.csv
%       Cell2
%           1996Cell2.csv
%           1997Cell2.csv
%           ...
%           2007Cell2.csv
%           Cell2SensorsGroups.csv
%       ...
%
%   Every cell has its own sub-folder, and every cell has a raw ASCII data
%   file (.csv) for each year covered, and every cell has a raw ASCII
```

```
%   SensorGroups file.
%
% o Function CellProperties must be accessible on the MATLAB path.
%
% Notes:
% o This routine is relatively long, but it is fast (i.e. 1-2 seconds).
%
% o This routine is designed so that it is general and stable, even as new
%   data, new sensors, and new cells are added.  However, as additional data
%   are added, this function will need to be rerun to make certain that the
%   information in 'Initialization.mat' is up to date.
%
% o The mutable function CellProperties, which is called by this
%   initialization function, will have to be updated as new cells are
%   added, but not this function.
%
% o This routine assigns "location group" numbers to sensors missing group
%   numbers in their associated SensorsGroups file.  This assignment is
%   important because some of the tests compare data within groups to
%   identify outliers.  Location group numbers are assigned to an ungrouped
%   sensor by finding the group with the median sensor depth that is
%   closest to the depth of the sensor.
%
% o This routine assigns a "tree" number to each sensor.  A tree is a
%   vertical collection of sensors at a single (station, offset).  The
%   trees are numbered 1, 2, ... for each cell.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%-----------------------------------------------------------------------------
```

A.2.4 CreateMatFiles.m

```
%-----------------------------------------------------------------------------
% usage: [] = CreateMatFiles()
%
%   A stand-alone function to convert the provided ASCII .csv files into
%   MATLAB-specific, compressed binary, .mat files.
%
% Arguments:
%   NONE
%
% Returns:
%   NONE
%
% Output:
%   A complete set of binary source files (i.e. Cellxxx.mat) in the source
%   folder identified in the initialization file.
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
```

```
%
% o The raw data files are in-place as a folder hierarchy off of the
%   RAW_ROOT folder. The raw data structure is organized like:
%
%   RAW_ROOT
%       Cell1
%           1996Cell1.csv
%           1997Cell1.csv
%           ...
%           2007Cell1.csv
%           Cell1SensorsGroups.csv
%       Cell2
%           1996Cell2.csv
%           1997Cell2.csv
%           ...
%           2007Cell2.csv
%           Cell2SensorsGroups.csv
%       ...
%
%   Every cell has its own sub-folder, and every cell has a raw ASCII data
%   file (.csv) for each year covered, and every cell has a raw ASCII
%   SensorGroups file.
%
% o If the SRC_ROOT folder does not exist, this function must have the
%   authority to create it.  This function must have the authority to write
%   to the SRC_ROOT folder.
%
% Notes:
% o This conversion was necessary for computational speed. Converting from
%   ASCII to MATLAB compressed binary increased the processing speed by
%   two orders of magnitude in some cases. Yes, the ASCII read and write
%   takes a lot of time.
%
% o An additional benefit of this ASCII to binary conversion is the
%   significant compression of the data files.  All of the ASCII files
%   combined require more than 4.45 GBytes, while the equivalent compressed
%   binary file require only 1.38 GBytes.
%
% o This function is specific to the data file structure given by MNDOT.
%
% o The raw source (.csv) files are found in sub-folders of the RAW_ROOT
%   folder, with one sub-folder per cell.  These sub-folders take names
%   of the form:  "Cell1", "Cell27", and "Cell93".
%
% o The raw source (.csv) files take names of the form "1998Cell1.csv",
%   "2001Cell27.csv", and "2004Cell93.csv".
%
% o The raw source (.csv) files have a single header line of the form:
%
%   "Cell,Day,Hour,Qhr,Minute,s_1,s_2,s_3,s_4,s_5,s_6,s_7,s_8,s_9,s_10,s_11"
%
%   "Cell,Day,Hour,Qhr,Minute,s_1,s_2,s_3,s_4,s_5,s_6,s_7,s_8,s_9,s_10,s_11,
%    s_12,s_13,s_14,s_15,s_16,s_17,s_18,s_19,s_20,s_21,s_22,s_23,s_24,s_25,
%    s_26,s_27,s_28,s_29,s_30,s_31,s_32,s_33,s_34,s_35,s_36,s_37,s_38,s_39,
```

```
%    s_40,s_41,s_42,s_43,s_44,s_45,s_46,s_47,s_48,s_49,s_50,s_51,s_52,s_53,
%    s_54,s_55,s_56,s_57,s_58,s_59,s_60,s_61,s_62,s_63,s_64,s_65,s_66,s_67,
%    s_68,s_69,s_70,s_71,s_72,s_73,s_74,s_75,s_76"
%
%   (without the " " marks).  All subsequent lines of the raw source files
%   contain one line for each measurement time, which are of the form:
%
%   "1,2002-1-1,6,2,35,-11.8765,-11.3189,-8.3722,-7.2619,-4.0752,-3.0087,
%    -1.3189,0.7480,2.8634,4.6692,8.9395"
%
%   "17,1996-1-1,0,0,5,-3,-2.90,-2.70,-2.80,0,0,0,0,0,0,0,-1.70,-1.60,-1.40,
%    -1.60,0,0,0,0,0,0,0,-2.40,-2.20,-2.10,-2.10,0,0,0,0,0,0,0,-1.90,-1.90,
%    -2,0,0,0,0,0,0,0,-1.90,-1.60,-1.50,-1.50,0,0,0,0,0,0,0,0.30,-1.30,-1.30,
%    2.60,0,0,0,0,0,0,0,-2,-1.90,-1.80,-2,0,0,0,0,0,0,0"
%
%   (again, without the " " marks).
%
%   These examples were taken from Cell1 and Cell17.  The number of sensors
%   may change from cell to cell.
%
% o The expectation of this format for the raw source files is built into
%   this function.  If the format changes, then this function will have to
%   be modified to reflect the changes.
%
% o This function produces one binary file for each source cell.  The
%   various years represented by the raw ASCII source files are combined
%   into one binary file.  Each of these binary files contains three
%   numeric arrays:
%
%       D   (Nx1) array of dates using MATLAB's datenum format,
%       T   (NxS) array of temperature data,
%       F   (NxS) array of uint16 flags.
%
%   where N is the total number of observations times for the cell (the sum
%   over all of the years), and S is the number of sensors in the cell.
%
% o In addition to the three numeric arrays, this function appends two
%   additional variables to each binary file: 'VERSION' and 'FLAGGED'.
%
%   'VERSION'   is of the form: "01-Jul-2009 06:14:12".  This is the
%               version number for the file:  the "datestr(now)" at which
%               the file was created.  Every time this file is modified by
%               one of the flagging functions this version number is
%               updated.
%
%   'FLAGGED'   is an unsigned 16-bit integer variable.  This is initialized
%               to "uint16(0)" in this function.  Every time this binary
%               file is modified by one of the flagging functions this
%               flagged variable is updated to reflect the union of all
%               flags applied so far.
%
% o The binary (.mat) files produced by this function are placed in the
%   SRC_ROOT folder.  If this folder does not exist it is created.
%
```

```
% o It may seem odd to designate the destination folder for this function’s
%   output as SRC_ROOT, but this function’s output is the source data for
%   all future processing.
%
% o This routine takes the "year/day/hour/minute" representation from the
%   ASCII source file and converts in into a MATLAB serial date number
%   (i.e. datenum).  These serial date numbers are in units of decimal days
%   since a nominal "00-Jan-0000".
%
% o The data in the generated Cellxxx.mat files are sorted into ascending
%   date order.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%------------------------------------------------------------------------
```

A.2.5 CreateSubsetRanges.m

```
%------------------------------------------------------------------------
% usage: [] = CreateSubsetRanges()
%
%   For each subset of similar sensors (type,group,location), compute the
%   median, lower quartile, and upper quartile temperature for every time
%   interval.
%
% Arguments:
%   NONE
%
% Returns:
%   NONE
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The binary source files (i.e. Cellxxx.mat) must exist in the source
%   folder identified in the initialization file.
%
% o The following flags have already been correctly applied:
%
%       FLAG_MISSING_DATA           missing data
%       FLAG_NOT_YET_OPERATIONAL    missing data at the beginning
%       FLAG_DEACTIVATED            missing data at the end
%       FLAG_TOO_SPARSE_DAY         not enough data in any day
%       FLAG_OUT_OF_RANGE           sensor outliers with annual & diurnal fit
%
% Notes:
% o These statistics are computed ignoring all data with one of the
%   following flags set:
%
%       FLAG_MISSING_DATA           missing data
%       FLAG_NOT_YET_OPERATIONAL    missing data at the beginning
```

```
%       FLAG_DEACTIVATED            missing data at the end
%       FLAG_TOO_SPARSE_DAY         not enough data in any day
%       FLAG_OUT_OF_RANGE           sensor outliers with annual & diurnal fit
%
%   This set of flags MUST be the same in all on the following functions:
%
%       CreateSubsetRange.m
%       Flag_PointExtremes.m
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%-------------------------------------------------------------------------
```

A.2.6 Flag_DailyExtremes.m

```
%-------------------------------------------------------------------------
% usage: [] = Flag_DailyExtremes()
%
%   For each sensor, identify all days in which
%
%   o   the daily minimum is too low
%   o   the daily minimum is too high
%
%   o   the daily maximum is too low
%   o   the daily maximum is too high
%
%   Flag all measurements in such inconsistent days.
%
% Arguments:
%   NONE
%
% Returns:
%   NONE
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The binary source files (i.e. Cellxxx.mat) must exist in the source
%   folder identified in the initialization file.
%
% o The following flags have already been correctly applied:
%
%       FLAG_MISSING_DATA           missing data
%       FLAG_NOT_YET_OPERATIONAL    missing data at the beginning
%       FLAG_DEACTIVATED            missing data at the end
%       FLAG_TOO_SPARSE_DAY         not enough data in any day
%       FLAG_OUT_OF_RANGE           sensor outliers with annual & diurnal fit
%
%   This set of flags MUST be the same in all on the following three
%   functions:
%
```

```
%       CreateDailyRange.m
%       Flag_DailyExtremes.m
%       Flag_DailyRange.m.
%
% Notes:
% o Too high and too low are determined by comparing a sensor's daily
%   extremes data to the daily extremes data of all similar sensors:  i.e.
%   sensors with the same type, group, and location.
%
% o These subset norms are computed month by month to allow for temporal
%   variable.
%
% o A day runs from midnight to midnight.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%-------------------------------------------------------------------------
```

A.2.7 Flag_DailyRange.m

```
%-------------------------------------------------------------------------
% usage: [] = Flag_DailyRange()
%
%   For each sensor, identify all days in which
%
%   o   the daily temperature range is too low
%   o   the daily temperature range is too high
%
%   Flag all measurements in such inconsistent days.
%
% Arguments:
%   NONE
%
% Returns:
%   NONE
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The binary source files (i.e. Cellxxx.mat) must exist in the source
%   folder identified in the initialization file.
%
% o The following flags have already been correctly applied:
%
%       FLAG_MISSING_DATA           missing data
%       FLAG_NOT_YET_OPERATIONAL    missing data at the beginning
%       FLAG_DEACTIVATED            missing data at the end
%       FLAG_TOO_SPARSE_DAY         not enough data in any day
%       FLAG_OUT_OF_RANGE           sensor outliers with annual & diurnal fit
%
%   This set of flags MUST be the same in all on the following three
```

```
%   functions:
%
%       CreateDailyRange.m
%       Flag_DailyExtremes.m
%       Flag_DailyRange.m.
%
% Notes:
% o Too high and too low are determined by comparing a sensor’s daily
%   range data to the daily range data of all similar sensors:  i.e.
%   sensors with the same type, group, and location.
%
% o These subset norms are computed month by month to allow for temporal
%   variable.
%
% o A day runs from midnight to midnight.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%-----------------------------------------------------------------------------
```

A.2.8 Flag_Deactivated.m

```
%-----------------------------------------------------------------------------
% usage: [] = Flag_Deactivated()
%
%   The Flag_Deactivated function flags all data after the last real
%   reading for each sensor.
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The binary source files (i.e. Cellxxx.mat) must exist in the source
%   folder identified in the initialization file.
%
% o The following flags have already been correctly applied:
%
%       FLAG_MISSING_DATA           missing data
%
% Notes:
% o It is anticipated that, under most circumstances, the subset of data
%   with <FLAG_DEACTIVATED> is a subset of the data with
%   <FLAG_MISSING_DATA>.  This flag is, nonetheless, useful since it gives
%   an explanation for the associated missing data.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%-----------------------------------------------------------------------------
```

A.2.9 Flag_InconsistentDay.m

```
%------------------------------------------------------------------------------
% usage: [] = Flag_InconsistentDay()
%
%   For each sensor, identify all days in which fewer than a specified
%   minimum number of data, or fewer than the specified minimum fraction
%   of the measurements, are good (unflagged) data.  Flag all measurements
%   in such inconsistent days.
%
% Arguments:
%   NONE
%
% Returns:
%   NONE
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The binary source files (i.e. Cellxxx.mat) must exist in the source
%   folder identified in the initialization file.
%
% o The following flags have already been correctly applied:
%
%       FLAG_MISSING_DATA            missing data
%       FLAG_NOT_YET_OPERATIONAL     missing data at the beginning
%       FLAG_DEACTIVATED             missing data at the end
%       FLAG_TOO_SPARSE_DAY          not enough data in any day
%
%       FLAG_OUT_OF_RANGE            sensor outliers with annual & diurnal fit
%       FLAG_NEIGHBORHOOD_OUTLIERS   sensor outliers with local neighborhood fit
%       FLAG_LAG_ONE_OUTLIERS        sensor outliers in lag one
%
%       FLAG_POINT_EXTREMES          subset outliers, record-by-record
%       FLAG_DAILY_RANGE             subset daily range outliers, day-by-day
%       FLAG_DAILY_EXTREMES          subset daily extreme outliers, day-by-day
%
%       FLAG_INTERMITTENT_DATA       too many flagged data around
%
% Notes:
% o The analysis for this flag is carried out on a sensor-by-sensor basis.
%   When identifying outliers for each sensor, only data from that specific
%   sensor's time series are used.
%
% o A day runs from midnight to midnight.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%------------------------------------------------------------------------------
```

A.2.10 Flag_InconsistentMonth.m

```
%-----------------------------------------------------------------------
% usage: [] = Flag_InconsistentMonth()
%
%   For each sensor, identify all months in which fewer than a specified
%   minimum number of data, or fewer than the specified minimum fraction
%   of the measurements, are good (unflagged) data.  Flag all measurements
%   in such inconsistent months.
%
% Arguments:
%   NONE
%
% Returns:
%   NONE
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The binary source files (i.e. Cellxxx.mat) must exist in the source
%   folder identified in the initialization file.
%
% o The following flags have already been correctly applied:
%
%       FLAG_MISSING_DATA            missing data
%       FLAG_NOT_YET_OPERATIONAL     missing data at the beginning
%       FLAG_DEACTIVATED             missing data at the end
%       FLAG_TOO_SPARSE_DAY          not enough data in any day
%
%       FLAG_OUT_OF_RANGE            sensor outliers with annual & diurnal fit
%       FLAG_NEIGHBORHOOD_OUTLIERS   sensor outliers with local neighborhood fit
%       FLAG_LAG_ONE_OUTLIERS        sensor outliers in lag one
%
%       FLAG_POINT_EXTREMES          subset outliers, record-by-record
%       FLAG_DAILY_RANGE             subset daily range outliers, day-by-day
%       FLAG_DAILY_EXTREMES          subset daily extreme outliers, day-by-day
%
%       FLAG_INTERMITTENT_DATA       too many flagged data around
%       FLAG_INCONSISTENT_DAY        too small a fraction of good data, day-by-day
%       FLAG_INCONSISTENT_WEEK       too small a fraction of good data, week-by-week
%
% Notes:
% o The analysis for this flag is carried out on a sensor-by-sensor basis.
%   When identifying outliers for each sensor, only data from that specific
%   sensor's time series are used.
%
% o A month runs from midnight on the morning of the first day of the
%   month, to midnight on the evening of the last day of the month.
%
% o This will "reflag" missing data.
%
% Version:
%   15 September 2009
```

```
%   MATLAB 7.8.0.347 (R2009a)
%--------------------------------------------------------------------------
```

A.2.11 Flag_InconsistentWeek.m

```
%--------------------------------------------------------------------------
% usage: [] = Flag_InconsistentWeek()
%
%   For each sensor, identify all weeks in which fewer than a specified
%   minimum number of data, or fewer than the specified minimum fraction
%   of the measurements, are good (unflagged) data.  Flag all measurements
%   in such inconsistent weeks.
%
% Arguments:
%   NONE
%
% Returns:
%   NONE
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The binary source files (i.e. Cellxxx.mat) must exist in the source
%   folder identified in the initialization file.
%
% o The following flags have already been correctly applied:
%
%       FLAG_MISSING_DATA            missing data
%       FLAG_NOT_YET_OPERATIONAL     missing data at the beginning
%       FLAG_DEACTIVATED             missing data at the end
%       FLAG_TOO_SPARSE_DAY          not enough data in any day
%
%       FLAG_OUT_OF_RANGE            sensor outliers with annual & diurnal fit
%       FLAG_NEIGHBORHOOD_OUTLIERS  sensor outliers with local neighborhood fit
%       FLAG_LAG_ONE_OUTLIERS        sensor outliers in lag one
%
%       FLAG_POINT_EXTREMES          subset outliers, record-by-record
%       FLAG_DAILY_RANGE             subset daily range outliers, day-by-day
%       FLAG_DAILY_EXTREMES          subset daily extreme outliers, day-by-day
%
%       FLAG_INTERMITTENT_DATA       too many flagged data around
%       FLAG_INCONSISTENT_DAY        too small a fraction of good data, day-by-day
%
% Notes:
% o The analysis for this flag is carried out on a sensor-by-sensor basis.
%   When identifying outliers for each sensor, only data from that specific
%   sensor's time series are used.
%
% o A week runs from midnight on Sunday morning to midnight on Saturday
%   night.
%
% o This will "reflag" missing data.
```

```
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%-------------------------------------------------------------------------
```

A.2.12 Flag_IntermittentData.m

```
%-------------------------------------------------------------------------
% usage: [] = Flag_IntermittentData()
%
%   For each sensor, this finds and flags data in sensors that are between
%   two flagged data points separated by less than XX days, where
%   XX is given by "MINIMUM_INTERMITTENT_TIME" in the initialization file.
%
% Arguments:
%   NONE
%
% Returns:
%   NONE
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The source files must exist in the designated folder and in the
%   anticipated format.
%
% o The following flags have already been correctly applied:
%
%       FLAG_MISSING_DATA            missing data
%       FLAG_NOT_YET_OPERATIONAL     missing data at the beginning
%       FLAG_DEACTIVATED             missing data at the end
%       FLAG_TOO_SPARSE_DAY          not enough data in any day
%
%       FLAG_OUT_OF_RANGE            sensor outliers with annual & diurnal fit
%       FLAG_NEIGHBORHOOD_OUTLIERS   sensor outliers with local neighborhood fit
%       FLAG_LAG_ONE_OUTLIERS        sensor outliers in lag one
%
%       FLAG_POINT_EXTREMES          subset outliers, record-by-record
%       FLAG_DAILY_RANGE             subset daily range outliers, day-by-day
%       FLAG_DAILY_EXTREMES          subset daily extreme outliers, day-by-day
%
% Notes:
% o The analysis for this flag is carried out on a sensor-by-sensor basis.
%   When identifying outliers for each sensor, only data from that specific
%   sensor’s time series are used.
%
% o A day runs from midnight to midnight.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
```

```
%------------------------------------------------------------------------------
```

A.2.13 Flag_LagOneOutliers.m

```
%------------------------------------------------------------------------------
% usage: [] = Flag_LagOneOutliers()
%
%   Identify and flag outliers in the lag one differences.
%
% Arguments:
%   NONE
%
% Returns:
%   NONE
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The source files must exist in the designated folder and in the
%   anticipated format.
%
% o The following flags have already been correctly applied:
%
%       FLAG_MISSING_DATA           missing data
%       FLAG_NOT_YET_OPERATIONAL    missing data at the beginning
%       FLAG_DEACTIVATED            missing data at the end
%       FLAG_TOO_SPARSE_DAY         not enough data in any day
%       FLAG_OUT_OF_RANGE           sensor outliers with annual & diurnal fit
%
% Notes:
% o The analysis for this flag is carried out on a sensor-by-sensor basis.
%   When identifying outliers for each sensor, only data from that specific
%   sensor's time series are used.
%
% o Outlier identification is then applied to the residuals.  The upper and
%   lower bounds for the residuals are defined as
%
%       Pr(r(i) < lowerbound) = PROBABILITY_OF_FALSE_POSITIVE/2
%       Pr(r(i) > upperbound) = PROBABILITY_OF_FALSE_POSITIVE/2
%
%   based upon a fitted Generalized Normal Distribution for the lag one
%   differences.
%
% o Any residual exceeding the upper or lower bounds are identified as
%   outliers and the associated data are flagged as out-of-range.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%------------------------------------------------------------------------------
```

A.2.14 Flag_MissingData.m

```
%-------------------------------------------------------------------------
% usage: [] = Flag_MissingData()
%
%   Identify and flag missing data (temperature fields) within the data
%   records present in all cells and all sensors.
%
% Arguments:
%   NONE
%
% Returns:
%   NONE
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The binary source files (i.e. Cellxxx.mat) must exist in the source
%   folder identified in the initialization file.
%
% Notes:
% o In the raw ASCII source files, missing data was designated with a value
%   of "0".  This was an unfortunate choice since 0 is a valid temperature
%   reading that is not uncommon in Minnesota.
%
% o From 1998 to the present, the temperature data is represented using
%   four or more digits to the right of the decimal point.  As such, it is
%   highly unlikely that a temperature reading of exactly 0.0000 is
%   collected. (None were found in the data analyzed.)  Thus, after 1998, a
%   value of exactly "0" is the raw data file can be reliably designated as
%   a missing value.
%
% o Prior to 1998, the temperature data is represented using at most two
%   digits to right of the decimal, and sometimes even these are truncated.
%   As such, some correct temperature measurements are encoded as "0",
%   which is indistinguishable from a missing value.
%
% o The reliable identification of missing data prior to 1998 requires
%   some analysis, and a few assumptions.  For example, if a sequence of
%   temperature measurements read
%
%       0.6, 0.4, 0.2, 0, -0.2, -0.4 ...
%
%   it is reasonable to believe that the "0" reading a  true measurement
%   and not a missing data value. On the other hand, if a sequence of
%   temperature measurements read
%
%       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
%
%   it is highly unlikely that the temperature would stay exactly equal to
%   0 for a long period of time, so these should be interpreted as a
%   sequence of missing data.
%
```

```
% o A recorded temperature value of "0" is flagged as missing if any one
%   of the following five conditions is true:
%
%   (1) year >= 1998;
%   (2) it is the first measurement of the first year;
%   (3) the previous measurement is missing;
%   (4) there are more than <MAX_RUN_AT_0> consecutive 0's.
%   (5) the previous measurement differs by more than <MAX_DELTA_T_AT_0>;
%
% o There are five reported counts for flagged data, which align with these
%   five conditions.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%-------------------------------------------------------------------------
```

A.2.15 Flag_NeighborhoodOutliers.m

```
%-------------------------------------------------------------------------
% usage: [] = Flag_NeighborhoodOutliers()
%
%   Identify and flag outliers on a sensor-by-sensor basis using a local
%   neighborhood (moving average) fit.
%
% Arguments:
%   NONE
%
% Returns:
%   NONE
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The source files must exist in the designated folder and in the
%   anticipated format.
%
%       FLAG_MISSING_DATA           missing data
%       FLAG_NOT_YET_OPERATIONAL    missing data at the beginning
%       FLAG_DEACTIVATED            missing data at the end
%       FLAG_TOO_SPARSE_DAY         not enough data in any day
%       FLAG_OUT_OF_RANGE           sensor outliers with annual & diurnal fit
%
% Notes:
% o The analysis for this flag is carried out on a sensor-by-sensor basis.
%   When identifying outliers for each sensor, only data from that specific
%   sensor's time series are used.
%
% o Only data that is NOT flagged with one of the following is considered
%   in this analysis:
%
%       FLAG_MISSING_DATA           missing data
```

```
%       FLAG_NOT_YET_OPERATIONAL    missing data at the beginning
%       FLAG_DEACTIVATED            missing data at the end
%       FLAG_TOO_SPARSE_DAY         not enough data in any day
%       FLAG_OUT_OF_RANGE           sensor outliers with annual & diurnal fit
%
% o For each sensor, the representative measurement spacing is taken as the
%   median measurement spacing:
%
%       DeltaD [day/measurement]
%
% o The NEIGHBORHOOD_WIDTH [days] is given in the initialization. The
%   number of measurements in a neighborhood, 2w, is then given by
%
%       2w            = NEIGHBORHOOD_WIDTH / DeltaD
%       [measurement] = [days]             / [day/measurement]
%
%   That is,
%
%       w = ceil( NEIGHBORHOOD_WIDTH / (2*DeltaD) );
%
% o The moving average value at each measurement is given by:
%
%                [  i+w       ]
%                [  SUM  T(j) ]  -  T(i)
%                [ j=i-w      ]
%       MA(i) =  -----------------------
%                          2w
%
%   where MA(i) is the moving average, and T(j) is the j'th temperature
%   measurement for the current sensor.
%
% o Note that the measurement associated with time i, T(i), is NOT included
%   in the moving average at time i, MA(i).  The moving average is the
%   neighborhood mean NOT including the point of interest.
%
% o Note also that the first w measurements and the last w measurements do
%   not have a full neighborhood, so they are not tested by this function.
%
% o The residual (prediction error) is computed as
%
%       r(i) = T(i) - MA(i)
%
% o Outlier identification is then applied to the residuals.  The upper and
%   lower bounds for the residuals are defined as
%
%       Pr(r(i) < lowerbound) = PROBABILITY_OF_FALSE_POSITIVE/2
%       Pr(r(i) > upperbound) = PROBABILITY_OF_FALSE_POSITIVE/2
%
%   based upon a fitted Generalized Normal Distribution for the lag one
%   differences.
%
% o Any residual exceeding the upper or lower bounds are identified as
%   outliers and the associated data are flagged as a neighborhood outlier.
%
```

```
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%-------------------------------------------------------------------------
```

A.2.16 Flag_NotYetOperational.m

```
%-------------------------------------------------------------------------
% usage: [] = Flag_NotYetOperational()
%
%   The Flag_NotYetOperational function flags all data until the first real
%   reading for each sensor.
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The binary source files (i.e. Cellxxx.mat) must exist in the source
%   folder identified in the initialization file.
%
% o The following flags have already been correctly applied:
%
%       FLAG_MISSING_DATA           missing data
%
% Notes:
% o It is anticipated that, under most circumstances, the subset of data
%   with <FLAG_NOT_YET_OPERATIONAL> is a subset of the data with
%   <FLAG_MISSING_DATA>.  This flag is, nonetheless, useful since it gives
%   an explanation for the associated missing data.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%-------------------------------------------------------------------------
```

A.2.17 Flag_OutOfRange.m

```
%-------------------------------------------------------------------------
% usage: [] = Flag_OutOfRange()
%
%   Identify and flag outliers on a sensor-by-sensor basis after removing
%   the annual and diurnal periodic variation.
%
% Arguments:
%   NONE
%
% Returns:
%   NONE
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
```

```
% o The source files must exist in the designated folder and in the
%   anticipated format.
%
% o The following flags have already been correctly applied:
%
%       FLAG_MISSING_DATA           missing data
%       FLAG_NOT_YET_OPERATIONAL    missing data at the beginning
%       FLAG_DEACTIVATED            missing data at the end
%       FLAG_TOO_SPARSE_DAY         not enough data in any day
%
% Notes:
% o The analysis for this flag is carried out on a sensor-by-sensor basis.
%   When identifying outliers for each sensor, only data from that specific
%   sensor's time series are used.
%
% o First, the periodic annual (seasonal) and diurnal (daily) temperature
%   variations are removed using a least-squares fit to the following
%   simple periodic model:
%
%       T(t) = To +
%               a1 cos( 2*pi*t/365.25 ) + b1 sin( 2*pi*t/365.25 ) +
%               a2 cos( 2*pi*t        ) + b2 sin( 2*pi*t        ) +
%               r(t)
%
%   where
%   -   t       is the time in days,
%   -   T(t)    is the temperature at time t,
%   -   To      is the fitted mean temperature for the entire time series,
%   -   a1,b1   are the fitted coefficients for the annual variation,
%   -   a2,b2   are the fitted coefficients for the diurnal variation, and
%   -   r(t)    is the residual at time t.
%
% o Nonparametric outlier identification is then applied to the residuals.
%   The upper and lower quartile spread for the residuals are defined by
%
%       upper_quartile_spread = 75 percentile - 50 percentile
%       lower_quartile_spread = 50 percentile - 25 percentile
%
%   The upper and lower bounds for the normal residuals are defined as
%
%       upper_bound = median + zeta*upper_quartile_spread
%       lower_bound = median + zeta*upper_quartile_spread
%
%   where zeta is defined such that
%
%       Pr(r(i) < lowerbound) = PROBABILITY_OF_FALSE_POSITIVE/2
%       Pr(r(i) > upperbound) = PROBABILITY_OF_FALSE_POSITIVE/2
%
%   based upon a fitted Normal Distribution for the residuals.
%
% o Any residual exceeding the upper or lower bounds are identified as
%   outliers and the associated data are flagged as out-of-range.
%
% o Cell 6, Sensor 7 is an excellent demonstration of the discerning power
```

```
%   of this test.  The test identifies the anomalous data in the summer of
%   2006, even though all of the data is within the range of normal annual
%   variation.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%-------------------------------------------------------------------------
```

A.2.18 Flag_PointExtremes.m

```
%-------------------------------------------------------------------------
% usage: [] = Flag_PointExtremes()
%
%   For each sensor, identify all temperature measurements that are
%
%   o   too low (relative to measurements taken at the same time with
%       similar sensors)
%
%   o   too high (relative to measurements taken at the same time with
%       similar sensors)
%
%   Flag all measurements in such inconsistent data.
%
% Arguments:
%   NONE
%
% Returns:
%   NONE
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The binary source files (i.e. Cellxxx.mat) must exist in the source
%   folder identified in the initialization file.
%
% o The following flags have already been correctly applied:
%
%       FLAG_MISSING_DATA            missing data
%       FLAG_NOT_YET_OPERATIONAL     missing data at the beginning
%       FLAG_DEACTIVATED             missing data at the end
%       FLAG_TOO_SPARSE_DAY          not enough data in any day
%       FLAG_OUT_OF_RANGE            sensor outliers with annual & diurnal fit
%
% Notes:
% o Too high and too low are determined by comparing a sensor's data to the
%   data of all similar sensors:  i.e. sensors with the same type, group,
%   and location.
%
% o Nonparametric outlier identification is then applied to the residuals.
%   The upper and lower quartile spread for the residuals are defined by
%
```

```
%       upper_quartile_spread = 75 percentile - 50 percentile
%       lower_quartile_spread = 50 percentile - 25 percentile
%
%   The upper and lower bounds for the normal residuals are defined as
%
%       upper_bound = median + zeta*upper_quartile_spread
%       lower_bound = median + zeta*upper_quartile_spread
%
%   where zeta is defined such that
%
%       Pr(r(i) < lowerbound) = PROBABILITY_OF_FALSE_POSITIVE/2
%       Pr(r(i) > upperbound) = PROBABILITY_OF_FALSE_POSITIVE/2
%
%   based upon a fitted Normal Distribution for the relative residuals.
%
% o Because the number of data in each subset at each time may be quite
%   small, there is an ad hoc lower limit on the effective upper and lower
%   quartile spread set at 1 degree.  This ad hoc limit will not let the
%   acceptable range shrink too much, even when many of the sensor readings
%   are virtually identical.
%
% o These statistics are computed ignoring all data with one of the
%   following flags set:
%
%       FLAG_MISSING_DATA           missing data
%       FLAG_NOT_YET_OPERATIONAL    missing data at the beginning
%       FLAG_DEACTIVATED            missing data at the end
%       FLAG_TOO_SPARSE_DAY         not enough data in any day
%       FLAG_OUT_OF_RANGE           sensor outliers with annual & diurnal fit
%
%   This set of flags MUST be the same in all on the following functions:
%
%       CreateSubsetRange.m
%       Flag_PointExtremes.m
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%-------------------------------------------------------------------------
```

A.2.19 Flag_TooSparseDay.m

```
%-------------------------------------------------------------------------
% usage: [] = Flag_TooSparseDay()
%
%   For each sensor, all records from days with fewer than a minimum
%   number of data are flagged.
%
% Arguments:
%   NONE
%
% Returns:
%   NONE
```

```
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The source files must exist in the designated folder and in the
%   anticipated format.
%
% o The following flags have already been correctly applied:
%
%       FLAG_MISSING_DATA           missing data
%       FLAG_NOT_YET_OPERATIONAL    missing data at the beginning
%       FLAG_DEACTIVATED            missing data at the end
%
% Notes:
% o The acceptable minimum number of data for a sensor is given by the
%   manifest constant
%
%                MINIMUM_DATA_PER_DAY
%
%   which is loaded from the 'Initialization.mat' file.  To change this value,
%   change and rerun the CreateInitialization.m function.
%
% o These data are not necessarily bad, but they are tainted by proximity
%   to bad or missing data.
%
% o A day runs from midnight to midnight.
%
% Version:
%   22 August 2009
%   MATLAB 7.8.0.347 (R2009a)
%---------------------------------------------------------------------------
```

A.2.20 GeneralizedNormalFit.m

```
%---------------------------------------------------------------------------
% Usage: [mu,alpha,beta,fval,flag] = GeneralizedNormalFit( X, verbose )
%
%   Fits a Generalized Normal Distribution to the data given in X using
%   the maximum likelihood method.
%
% Arguments:
%   X       a vector of data.  This may be a row or a column vector, but
%           not a matrix.
%
%   verbose controls the presentation of intermediate results.
%           0   Silent running.
%           1   small iteration results.
%           2   detailed iteration results.
%           3   detailed iteration results with graphics.
%
% Returns:
%   mu      location parameter (real)
```

```
%   alpha   scale parameter (positive, real)
%   beta    shape parameter (positive, real)
%
%   fval    negative log-likelihood function at termination
%
%   flag    identifies the reason the algorithm terminated.
%           0   Number of iterations exceeded options. FAILURE.
%           1   Magnitude of gradient smaller than the TolFun tolerance.
%           2   Change in x was smaller than the TolX tolerance.
%           3   Change in the objective function value was less than the
%               TolFun tolerance.
%           5   Predicted decrease in the objective function was less than
%               the TolFun tolerance.
%
% Notes:
% o This function uses the built-in MATLAB "fminsearch" function to do the
%   actual minimization.
%
% o For more information on the Generalized Normal Distribution see:
%   http://en.wikipedia.org/wiki/Generalized_normal_distribution
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%-------------------------------------------------------------------------
```

A.2.21 PlotFlags.m

```
%-------------------------------------------------------------------------
% usage: [] = PlotFlags(starting_cell, starting_sensor)
%
%   Graphically preview the flag data in a continuous manner starting at
%   the specified cell and sensor.
%
% Arguments:
%   starting_cell    starting cell number (default = 1),
%   starting_sensor  starting sensor number in the cell (default = 1).
%
% Returns:
%   NONE
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The source files must exist in the designated folder and in the
%   anticipated format.
%
% Notes:
%
% o Press any key (e.g. <spacebar>) to step through the requested plots.
%
% o Press <Ctrl-C> to stop execution.
```

```
%
% o Red O are plotted for each set flag bit.  Blue . are plotted for each
%   unflagged record.
%
% o The plot title includes:
%   -   Cell #
%   -   Sensor #
%   -   Subset(type, group, loc, tree)
%       >   type: HMA, PCC, Other
%       >   group: 1,2, ..., 13
%       >   loc: ML=mainline, LV=low volume road
%       >   tree: tree are numbered within the cell
%   -   Unflagged(PercentActiveGood, PercentTotalGood)
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%---------------------------------------------------------------------------
```

A.2.22 PlotTemperatures.m

```
%---------------------------------------------------------------------------
% usage: [] = PlotTemperatures(starting_cell, starting_sensor, flag)
%
%   Graphically preview the temperature data in a continuous manner
%   starting at the specified cell and sensor.
%
% Arguments:
%   starting_cell    starting cell number (default = 1),
%   starting_sensor  starting sensor number in the cell (default = 1),
%   flag             flag number to highlight in red (default = 1).
%
% Returns:
%   NONE
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The source files must exist in the designated folder and in the
%   anticipated format.
%
% Notes:
% o The generated figure highlights the flagged data in green and the
%   specified data in red.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%---------------------------------------------------------------------------
```

A.2.23 `ResetAllFlags.m`

```
%-----------------------------------------------------------------------
% usage: [] = ResetAllFlags()
%
%   Resets ALL the flags in ALL cells.
%
% Arguments:
%   NONE
%
% Returns:
%   NONE
%
% Notes:
% o This function is NOT part of the normal data flow.  It is supplied as
%   a tool for exceptional circumstances.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%-----------------------------------------------------------------------
```

A.2.24 `ResetFlags.m`

```
%-----------------------------------------------------------------------
% usage: [] = ResetFlags( flags )
%
%   Resets the specified flags in ALL cells.
%
% Arguments:
%   flags       a vector of flag indices to reset back to 0.
%
% Returns:
%   NONE
%
% Notes:
% o This function is NOT part of the normal data flow.  It is supplied as
%   a tool for exceptional circumstances.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%-----------------------------------------------------------------------
```

A.3 Statistics Functions

A.3.1 `CreateGradientStats.m`

```
%-----------------------------------------------------------------------
% usage: [] = CreateGradientStats()
%
% The CreateGradientStats function goes through all the cells and
```

```
% calculates a table full of summary statistics about the sensor’s temporal
% gradient.  These statistics are calculated from a table of lag-one values
% for each sensor.
%
% The outputs are returned in the form of a .mat file named
%
%   ’GradientStats.mat’
%
% that is saved in the Data folder. These statistics, in order of table
% columns, are:
%
%   Cell Number
%   Pavetype
%   Sensor Number
%   Depth of Sensor (feet)
%   Count
%   Average
%   Variance
%   Standard Deviation
%   Coefficient of Skewness
%   Minimum
%   Maximum
%   Range
%   Median
%   Lower Quartile
%   Upper Quartile
%   Inter Quartile Range
%   1st Percentile
%   5th Percentile
%   10th Percentile
%   90th Percentile
%   95th Percentile
%   99th Percentile
%   Median Absolute Deviation
%   Trimmean
%   Layer Thickness
%   Flag Count
%   NaN Count
%
% Arguments:
%   NONE
%
% Returns:
%   CreateGradientStats.mat in the SRC_ROOT folder.
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The binary source files (i.e. Cellxxx.mat) must exist in the source
%   folder identified in the initialization file.
%
% o All flags have been correctly applied.
%
```

```
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%---------------------------------------------------------------------------
```

A.3.2 CreateSensorStats.m

```
%---------------------------------------------------------------------------
% usage: [] = CreateSensorStats()
%
% The CreateSensorStats function goes through all the cells and calculates
% a table full of summary statistics about the sensor.
%
% These outputs are returned in the form of a .mat file named
%
%   'SensorStats.mat'
%
% that is saved in the Data folder. These statistics in order of table
% column are:
%
%   Cell Number
%   Pavetype
%   Sensor Number
%   Depth of Sensor (feet)
%   Count
%   Average
%   Variance
%   Standard Deviation
%   Coefficient of Skewness
%   Minimum
%   Maximum
%   Range
%   Median
%   Lower Quartile
%   Upper Quartile
%   Inter Quartile Range
%   1st Percentile
%   5th Percentile
%   10th Percentile
%   90th Percentile
%   95th Percentile
%   99th Percentile
%   Median Absolute Deviation
%   Trimmean
%   Layer Thickness
%   Flag Count
%   NaN Count
%   NYO Count
%   Deactivated Count
%   Range Error Count
%   Intermittency Count
%   Supergroup Error Count
%   Lag One Error Count
```

```
%
% Arguments:
%   NONE
%
% Returns:
%   CreateSensorStats.mat in the SRC_ROOT folder.
%
% Preconditions:
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The binary source files (i.e. Cellxxx.mat) must exist in the source
%   folder identified in the initialization file.
%
% o All flags have been correctly applied.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%-------------------------------------------------------------------------
```

A.3.3 DepthStatCall.m

```
%-------------------------------------------------------------------------
% usage: [] = DepthStatCall(C, min_depth, max_depth)
%
%       Example: DepthStatCall(4, 1, 3.5)
%
% The DepthStatCall function takes the arguments of cell number and range
% of sensor depths (in inches) and outputs a summary of the sensor
% statistics for all sensors in that cell that fall within the specified
% depth range.
%
%   Cell Number
%   Pavetype
%   Sensor Number
%   Depth of Sensor (feet)
%   Count
%   Average
%   Variance
%   Standard Deviation
%   Coefficient of Skewness
%   Minimum
%   Maximum
%   Range
%   Median
%   Lower Quartile
%   Upper Quartile
%   Inter Quartile Range
%   1st Percentile
%   5th Percentile
%   10th Percentile
%   90th Percentile
```

```
%   95th Percentile
%   99th Percentile
%   Median Absolute Deviation
%   Trimmean
%   Layer Thickness
%   Flag Count
%   NaN Count
%   NYO Count
%   Deactivated Count
%   Range Error Count
%   Intermittency Count
%   SuperGroup Error Count
%   Lag One Error Count
%
% Arguments:
%   o Cell number
%   o Minimum sensor depth (in)
%   o Maximum sensor depth (in)
%
% Returns:
%   NONE
%
% Preconditions:
%
% o CreateSensorStats must be run prior to using this driver function.
%
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The binary source files (i.e. Cellxxx.mat) must exist in the source
%   folder identified in the initialization file.
%
% o All flags have been correctly applied.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%---------------------------------------------------------------------------
```

A.3.4 GradientStatCall.m

```
%---------------------------------------------------------------------------
% usage: [] = GradientStatCall(C, S)
%
%       Example: GradientStatCall(4, 9)
%
% The GradientStatCall function is a driver funtion for the GradientStats
% routine. Cell number and sensor number are input, and the outputs are
% arranged as follows:
%
%   Cell Number
%   Pavetype
%   Sensor Number
```

```
%   Depth of Sensor (feet)
%   Count
%   Average
%   Variance
%   Standard Deviation
%   Coefficient of Skewness
%   Minimum
%   Maximum
%   Range
%   Median
%   Lower Quartile
%   Upper Quartile
%   Inter Quartile Range
%   1st Percentile
%   5th Percentile
%   10th Percentile
%   90th Percentile
%   95th Percentile
%   99th Percentile
%   Median Absolute Deviation
%   Trimmean
%   Layer Thickness
%   Flag Count
%   NaN Count
%
% Arguments:
%   o Cell number
%   o Sensor number
%
% Returns:
%   NONE
%
% Preconditions:
%
% o CreateGradientStats must be run prior to using this driver function.
%
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The binary source files (i.e. Cellxxx.mat) must exist in the source
%   folder identified in the initialization file.
%
% o All flags have been correctly applied.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%
% Project:
%        Minnesota Department of Transportation
%        Mn/ROAD Data Mining, Evaluation and Qualification Phase 1
%
% Written by:
%   Dr. Randal J. Barnes and his research group
```

```
%   Department of Civil Engineering
%   Universtiy of Minnesota
%---------------------------------------------------------------------------
```

A.3.5 SensorStatCall.m

```
%---------------------------------------------------------------------------
% usage: [] = SensorStatCall(C, S)
%
%       Example: SensorStatCall(4, 9)
%
% The SensorStatCall function takes the arguments of cell number and sensor
% number (corresponding to the sensor’s column in the cell’s T matrix) and
% outputs a summary of the sensor’s statistics.
%
%   Cell Number
%   Pavetype
%   Sensor Number
%   Depth of Sensor (feet)
%   Count
%   Average
%   Variance
%   Standard Deviation
%   Coefficient of Skewness
%   Minimum
%   Maximum
%   Range
%   Median
%   Lower Quartile
%   Upper Quartile
%   Inter Quartile Range
%   1st Percentile
%   5th Percentile
%   10th Percentile
%   90th Percentile
%   95th Percentile
%   99th Percentile
%   Median Absolute Deviation
%   Trimmean
%   Layer Thickness
%   Flag Count
%   NaN Count
%   NYO Count
%   Deactivated Count
%   Range Error Count
%   Intermittency Count
%   Supergroup Error Count
%   Lag One Error Count
%
% Arguments:
%   o Cell number
%   o Sensor number
%
```

```
% Returns:
%   NONE
%
% Preconditions:
%
% o CreateSensorStats must be run prior to using this driver function.
%
% o An up-to-date "Initialization.mat" file must exist in the current
%   working directory.
%
% o The binary source files (i.e. Cellxxx.mat) must exist in the source
%   folder identified in the initialization file.
%
% o All flags have been correctly applied.
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%
% Project:
%        Minnesota Department of Transportation
%        Mn/ROAD Data Mining, Evaluation and Qualification Phase 1
%
% Written by:
%   Dr. Randal J. Barnes and his research group
%   Department of Civil Engineering
%   Universtiy of Minnesota
%--------------------------------------------------------------------------
```

A.4 Created Data Files

A.4.1 `Cell###.mat` Files

Origin

There is one cell `.mat` file of the form `Cell###.mat` created for each Cell, where the “`###`” identifies the cell number. For example, the cell `.mat` file for Cell #1 is `Cell001.mat`. These cell `.mat` files are created by the function `CreateMatFiles.m`. The cell `.mat` files are compressed binary files.

Purpose

The `Cell###.mat` files contain all of the raw data and the associated flags, in a standard, and rapidly accessible, compressed binary format. These files are used by **all** of the computational and reporting components in this system.

Contents

The cell `.mat` files contain all[1] of the time, temperature, and flag information for the associated cell. Each cell `.mat` file contains three large arrays:

- D: ($N \times 1$) array of dates using MATLAB's `datenum` format,
- T: ($N \times S$) array of temperature data, and
- F: ($N \times S$) array of 16-bit unsigned integer flags.

where N is the total number of observations for the cell (the sum over all of the years) and S is the number of sensors in the cell.

The cell `.mat` files also contain two additional variables:

`FLAGGED` A scalar variable tracking which flags have been applied.

`VERSION` A string variable tracking when the file was created. The version number is the date of creation.

A.4.2 `DailyRanges.mat` File

Origin

There is one `DailyRanges.mat` file for all of the cells combined. This file is created by the function `CreateDailyRanges.m`. The `DailyRanges.mat` file is in a compressed binary format.

Purpose

The `DailyRanges.mat` file is created to serve as a rapidly accessible source of information for the statistically-based acceptable ranges. This information is used in the following functions:

- `Flag_DailyExtremes.m`
- `Flag_DailyRange.m`

The information contained in the `DailyRanges.mat` files is computationally intensive to create and, as such, can not be recreated each time it is needed. This file exists for computational speed only.

[1]This includes all years and all sensors in the cell.

Contents

The `DailyRanges.mat` file contains one very large[2] array: R. Each row of the R array is associated with a single day and a single sensor. The columns of the R array are

- R(:,1) integer day (as a `datenum`)
- R(:,2) number of data in this range
- R(:,3) minimum temperature for the day
- R(:,4) maximum temperature for the day
- R(:,5) temperature range for the day
- R(:,6) cell number
- R(:,7) sensor number
- R(:,8) cell type (1=HMS, 2=PCC, 3=other)
- R(:,9) sensor group (depth group 1, 2, ..., 13)
- R(:,10) cell location (1 = Mainline, 2 = Low Volume Road)
- R(:,11) month

To assist in keeping the columns straight, even if the contents changes over time, the following manifest constants are defined in this routine and sorted in the `DailyRange.mat` file.

`R_DATE` = 1; % integer day

`R_COUNT` = 2; % # of data in this range

`R_MIN` = 3; % daily minimum

`R_MAX` = 4; % daily maximum

`R_RANGE` = 5; % daily range

`R_CELLNO` = 6; % cell number

`R_SENSOR` = 7; % sensor number

`R_TYPE` = 8; % cell's pavement type [1=HMA, 2=PCC, 3=Other]

`R_GROUP` = 9; % sensor's depth group [1 ... 13]

`R_LOC` = 10; % cell's location [1=ML, 2=LVR]

`R_MONTH` = 11; % month [1 ... 12]

The `DailyRange.mat` file contains an additional variable:

`VERSION` A string variable tracking when the file was created. The version number is the date of creation.

[2]For example, the R array generated for the data consider in this project contains $3,601,780$ rows.

A.4.3 `SubsetRanges.mat` File

Origin

There is one `SubsetRanges.mat` file for all of the cells combined. This file is created by the function `CreateSubsetRanges.m`. The `SubsetRanges.mat` file is in a compressed binary format.

Purpose

The `SubsetRanges.mat` file is created to serve as a rapidly accessible source of information for the statistically-based acceptable ranges. This information is used in the function:

- `Flag_PointExtremes.m`

The information contained in the `SubsetRanges.mat` files is computationally intensive to create and, as such, can not be recreated each time it is needed. This file exists for computational speed only.

Contents

The `SubsetRanges.mat` file contains four very large 4D arrays. Each of the four arrays is indexed by

- `type`: pavement type, with 1=HMA, 2=PCC, and 3=other
- `group`: depth group, which includes values in the range $\{1, 2, \ldots, 13\}$
- `loc`: cell location, with 1 = Mainline, and 2 = Low Volume Road
- `tim`: time increment index, which includes values in the range $\{1, 2, \ldots, \mathsf{TIME\_COUNT}\}$,

where

```
TIME_COUNT = ceil( (END_DATE - START_DATE + 1)/TIME_INTERVAL )
```

The time increment indexes the `TIME_INTERVAL` intervals[3] through the complete time history of the data. This baroque construction was necessary since to manage the observations that were taken a similar, but not exactly equal, times.

For the considered data, $\mathsf{TIME\_COUNT} = 420,864$. Thus, each of the four arrays contains

$$3 \times 13 \times 2 \times 420,864 = 32,827,392 \qquad \text{(A.1)}$$

total elements.

The four very large 4D arrays in the `SubsetRanges.mat` file are

- `GCNT` $(3 \times 13 \times 2 \times$ `TIME_COUNT`$)$ array of counts of temperature measurements taken in this subset and time increment.

[3]The manifest constant `TIME_INTERVAL` is set to 15 minutes in the function `CreateInitialization.m`.

- GLQS ($3 \times 13 \times 2 \times$ TIME_COUNT) array of lower quartile spreads[4] of temperature measurements taken in this subset and time increment.
- GMED ($3 \times 13 \times 2 \times$ TIME_COUNT) array of medians[5] of temperature measurements taken in this subset and time increment.
- GUQS ($3 \times 13 \times 2 \times$ TIME_COUNT) array of upper quartile spreads[6] of temperature measurements taken in this subset and time increment.

A.4.4 GradientStats.mat File

Origin

There is one GradientStats.mat file for all of the cells and all of the sensors combined. This file is created by the function CreateGradientStats.m. The GradientStats.mat file is in a compressed binary format.

Purpose

The GradientStats.mat file is created to serve as a rapidly accessible source of statistics for post-processing and review. The information contained in the GradientStats.mat files is computationally intensive to create and, as such, can not be recreated each time it is needed. This file exists for computational speed only.

Contents

The GradientStats.mat file contains one large array: gradient_stats. Each row of the array is associated with a single sensor. The columns of the array include:

- Cell Number
- Pavetype
- Sensor Number
- Depth of Sensor (feet)
- Count
- Average
- Variance
- Standard Deviation
- Coefficient of Skewness

[4]See Section 5.2.5 for a discussion on the lower quartile spread.
[5]See Section 5.2.1 for a discussion on the median.
[6]See Section 5.2.6 for a discussion on the upper quartile spread.

- Minimum
- Maximum
- Range
- Median
- Lower Quartile
- Upper Quartile
- Inter Quartile Range
- 1st Percentile
- 5th Percentile
- 10th Percentile
- 90th Percentile
- 95th Percentile
- 99th Percentile
- Median Absolute Deviation
- Trimmean
- Layer Thickness
- Flag Count
- NaN Count

These are summary statistics about the each sensor's temporal gradient. These statistics are calculated from a table of lag-one values for each sensor.

A.4.5 `SensorStats.mat` File

Origin

There is one `SensorStats.mat` file for all of the cells and all of the sensors combined. This file is created by the function `CreateSensorStats.m`. The `SensorStats.mat` file is in a compressed binary format.

Purpose

The `SensorStats.mat` file is created to serve as a rapidly accessible source of statistics for post-processing and review. The information contained in the `SensorStats.mat` files is computationally intensive to create and, as such, can not be recreated each time it is needed. This file exists for computational speed only.

Contents

The `SensorStats.mat` file contains one large array: `sensor_stats`. Each row of the array is associated with a single sensor. The columns of the array include:

- Cell Number
- Pavetype
- Sensor Number
- Depth of Sensor (feet)
- Count
- Average
- Variance
- Standard Deviation
- Coefficient of Skewness
- Minimum
- Maximum
- Range
- Median
- Lower Quartile
- Upper Quartile
- Inter Quartile Range
- 1st Percentile
- 5th Percentile
- 10th Percentile
- 90th Percentile
- 95th Percentile
- 99th Percentile
- Median Absolute Deviation
- Trimmean

- Layer Thickness
- Flag Count
- NaN Count
- NYO Count
- Deactivated Count
- Range Error Count
- Intermittency Count
- Supergroup Error Count
- Lag One Error Count

These are summary statistics about the each sensor's temperature measurements. These qualitative temperature statistics are calculated using only the unflagged data records.

Appendix B

Editable Constants

The following is extracted from the `CreateInitialization.m` function. This function defines all of the manifest constants used throughout the complete package.

```
%----------------------------------------------------------------------------
% In this section we define constants for each of the flags.
%----------------------------------------------------------------------------

% Missing data flags
FLAG_MISSING_DATA           = 1;    % missing data
FLAG_NOT_YET_OPERATIONAL    = 2;    % missing data at the beginning
FLAG_DEACTIVATED            = 3;    % missing data at the end
FLAG_TOO_SPARSE_DAY         = 4;    % not enough data in any day

% Time-series based
FLAG_OUT_OF_RANGE           = 5;    % sensor outliers with annual & diurnal fit
FLAG_NEIGHBORHOOD_OUTLIERS  = 6;    % sensor outliers with local neighborhood fit
FLAG_LAG_ONE_OUTLIERS       = 7;    % sensor outliers in lag one

% Subset-based flags
FLAG_POINT_EXTREMES         = 8;    % subset outliers, record-by-record
FLAG_DAILY_RANGE            = 9;    % subset daily range outliers, day-by-day
FLAG_DAILY_EXTREMES         = 10;   % subset daily extreme outliers, day-by-day

% Sensor-by-sensor consistency
FLAG_INTERMITTENT_DATA      = 11;   % too many flagged data points around
FLAG_INCONSISTENT_DAY       = 12;   % too few good data, day-by-day
FLAG_INCONSISTENT_WEEK      = 13;   % too few good data, week-by-week
FLAG_INCONSISTENT_MONTH     = 14;   % too few good data, month-by-month

NUMBER_OF_FLAGS             = 14;   % Current number of flags.

%----------------------------------------------------------------------------
% In this section we define the scaling and calibration constants for each
% of the tests.
%----------------------------------------------------------------------------

%............................
% The maximum allowable temperature change around 0 to distinguish between
% missing data and a true temperature of 0. This is for data prior to 1998
% only.  See Flag_MissingData.m for a more detailed explanation.
%
% Used in:
% o Flag_MissingData.m
%............................
MAXIMUM_DELTA_T_AT_0 = 2;

%............................
% The maximum number of consecutive 0's before flagging all of them as
% missing data. This is for data prior to 1998 only.  See Flag_MissingData.m
% for a more detailed explanation.
%
```

```
% Used in:
% o Flag_MissingData.m
%............................
MAXIMUM_RUN_AT_0 = 24;

%............................
% The acceptable minimum number of data for a sensor in any day.  All records
% from days with fewer than this minimum number of data are flagged.
%
% Used in:
% o Flag_TooSparseDay.m
%............................
MINIMUM_DATA_PER_DAY = 48;

%............................
% The minimum number of data to generate reliable order statistics (quartiles)
% from a general collection of data.
%
% Used in:
% o CreateSubsetRanges.m
% o Flag_DailyExtremes.m
% o Flag_DailyRange.m
% o Flag_LagOneOutliers.m
% o Flag_NeighborhoodOutliers.m
% o Flag_OutOfRange.m
% o Flag_PointExtremes.m
%............................
MINIMUM_DATA_FOR_STATS = 48;

%............................
% The target probability for a false positive in the various statistically-
% based tests.  That is, the probability of flagging a datum as an outlier
% when in fact is merely an extreme value in the population.  This is
% comparable to the "significance" (alpha) in classical statistical hypothesis
% testing.
%
% Used in:
% o Flag_DailyExtremes.m
% o Flag_DailyRange.m
% o Flag_LagOneOutliers.m
% o Flag_NeighborhoodOutliers.m
% o Flag_OutOfRange.m
% o Flag_PointExtremes.m
%............................
PROBABILITY_OF_FALSE_POSITIVE = 0.00001;   % 1 in 100,000

%............................
% The width of the time neighborhood in the computation of the moving average.
% This is given in units of [day].  This width is the entire duration of the
% time window -- that is, NEIGHBORHOOD_WIDTH/2 before and NEIGHBORHOOD_WIDTH/2
% after.
%
% Used in:
% o Flag_NeighborhoodOutliers.m
```

```
%............................
NEIGHBORHOOD_WIDTH = 3.0/24;   % [day] so 3.0/24 is 3 hours.

%............................
% The width of the time interval [day] used to determine when measurements are
% taken at the same nominal time.  If all measurements were taken at exactly
% the same time, and on exactly even intervals, this manifest constant would
% not be necessary.
%
% Used in:
% o CreateSubsetRanges.m
% o Flag_PointExtremes.m
%............................
TIME_INTERVAL = 0.25/24;   % [day] so 0.25/24 is 15 minutes.

%............................
% The maximum time separation [day] between one measurement and the next when
% computing the lag one statistics.
%
% Used in:
% o Flag_LagOneOutliers.m
%............................
MAX_LAG_ONE_DELTA_T = 0.50/24;   % [day] so 0.50/25 is 30 minutes.

%............................
% The minimum number of data in a subset of similar sensors at a specific
% time to make reliable comparisons.  See Flag_PointExtremes.m for more
% details.
%
% Used in:
% o Flag_PointExtremes.m
%............................
MINIMUM_DATA_IN_SUBSET = 10;

%............................
% The mimimum duration [day] of a sequence of good (unflagged) measurements
% for the data to be considered reliable.
%
% Used in:
% o Flag_IntermittentData.m
%............................
MINIMUM_INTERMITTENT_TIME = 6/24;   % [day] so 6/24 is 6 hours.

%............................
% If any given sensor has fewer than this many data in any given day, then
% all data collected by the given sensor in the given day are considered
% unreliable.
%
% If the fraction of good data (unflagged) from any given sensor in any given
% day is less than this fraction, then all data collected by the given sensor
% in the given day are considered unreliable.
%
% Used in:
% o Flag_InconsistentDay.m
```

```
%............................
MINIMUM_GOOD_DATA_DAY              = 10;
MINIMUM_FRACTION_GOOD_DATA_DAY     = 0.5;

%............................
% If any given sensor has fewer than this many data in any given week, then
% all data collected by the given sensor in the given week are considered
% unreliable.
%
% If the fraction of good data (unflagged) from any given sensor in any given
% week is less than this fraction, then all data collected by the given sensor
% in the given week are considered unreliable.
%
% Used in:
% o Flag_InconsistentWeek.m
%............................
MINIMUM_GOOD_DATA_WEEK             = 70;
MINIMUM_FRACTION_GOOD_DATA_WEEK    = 0.5;

%............................
% If any given sensor has fewer than this many data in any given month, then
% all data collected by the given sensor in the given month are considered
% unreliable.
%
% If the fraction of good data (unflagged) from any given sensor in any given
% month is less than this fraction, then all data collected by the given sensor
% in the given month are considered unreliable.
%
% Used in:
% o Flag_InconsistentMonth.m
%............................
MINIMUM_GOOD_DATA_MONTH            = 280;
MINIMUM_FRACTION_GOOD_DATA_MONTH   = 0.5;
```

Appendix C

Future Expansion: Adding New Cells

C.1 Background

All material properties for cells considered throughout this entire system are determined by the helper function `CellProperties.m`. Thus, as additional cells are added to the system, this function **must** be updated. In designing this system the mutable information was partitioned from the automated initialization. The changing parts (e.g. new cells) are defined in this function, while the more static components are set in the `CreateInitialization.m` function (see Appendix B).

> It is anticipated that the function `CellProperties.m` will be replaced by a more generic routine that automatically ties into the standard data base. In any event, as new cells are added, this function must be expanded.

C.2 Minimum Requirements

A call to the function `CellProperties.m` takes the form

```
[type, loc, mat, thick] = CellProperties(cellno, depth);
```

where the input arguments are

- `cellno`: cell number of interest.
- `depth`: depth [ft] of interest.

The arguments `cellno` and `depth` may be scalars, vectors, or matrices. A scalar input for either of the two arguments is interpreted as a constant matrix, equal to the input value, and of the same size as the other input argument. If both arguments are non-scalars then they must be of equal size.

The returned values are

- `type`: pavement type with 0=cell not found, 1=HMA, 2=PCC, 3=Other.
- `loc`: cell location with 1=ML, 2=LVR.
- `mat`: material type, with a 3-letter abbreviation[1] for material type at the requested depth.
- `thick`: designed thickness [ft] of the layer at the requested depth.

If the input cell number is not found, the return values are `[0, 0, ' ', 0]`. The pavement type = 0 should be used to identify an out-of-bounds input.

[1]These specific abbreviations are not used for computational purposes, but they are are used to create more informative output. Thus, they can be changed without adverse consequences.

Appendix D

Example Run

D.1 Driver Function

The following is extracted from the file A_RunAll.m. This file is a driver function that executes the entire filtering/flagging system and builds the resulting completed data files. On the test platform[1] this run required a total elapsed time of 10510.4 seconds (approximately 3 hours).

```
function [] = A_RunAll()
%--------------------------------------------------------------------------
% usage: [] = A_RunAll()
%
% Version:
%   15 September 2009
%   MATLAB 7.8.0.347 (R2009a)
%
% Project:
%        Minnesota Department of Transportation
%        Mn/ROAD Data Mining, Evaluation and Qualification Phase 1
%
% Written by:
%   Dr. Randal J. Barnes and his research group
%   Department of Civil Engineering
%   University of Minnesota
%--------------------------------------------------------------------------
start_time = tic;

fprintf(1,'=========================================================\n');
fprintf(1,' RUN ALL                                                 \n');
fprintf(1,'                                                         \n');
fprintf(1,' Version:                                                \n');
fprintf(1,'     1 September 2009                                    \n');
fprintf(1,'     MATLAB 7.8.0.347 (R2009a)                           \n');
fprintf(1,'=========================================================\n');

CreateInitialization( '../Raw Data Oct 07', '../Data' );
CreateMatFiles;

Flag_MissingData;
Flag_NotYetOperational;
Flag_Deactivated;
Flag_TooSparseDay;
Flag_OutOfRange;

Flag_NeighborhoodOutliers;
Flag_LagOneOutliers;

CreateSubsetRanges;
Flag_PointExtremes;

CreateDailyRanges;
```

[1] The test platform was a Dell Studio XPS 435MT, with an Intel Core i7 CPU 920 running at 2.67 GHz, 12.0 GB of random access memory, and the Microsoft Vista Ultimate 64 operating system.

```
Flag_DailyRange;
Flag_DailyExtremes;

Flag_IntermittentData;
Flag_InconsistentDay;
Flag_InconsistentWeek;
Flag_InconsistentMonth;

fprintf(1,'================================================================\n');
toc( start_time );
```

D.2 Partial Output

The following is an abridged[2] copy of the diary file[3] generated during the run of `A_RunAll.m`.

```
A_RunAll
===========================================================
 RUN ALL

 Version:
     1 September 2009
     MATLAB 7.8.0.347 (R2009a)
===========================================================

06-Sep-2009 18:27:59: CreateInitialization
There are no files or folders matching "../Raw Data Oct 07/Cell61/*Cell61.csv".
    Cell 61 will be ignored.
There are no files or folders matching "../Raw Data Oct 07/Cell63/*Cell63.csv".
    Cell 63 will be ignored.
Created: Initialization.mat
Elapsed time is 0.733708 seconds.

06-Sep-2009 18:28:00: CreateMatFiles
Processing Cell: 1
        Source file: ../Raw Data Oct 07/Cell1/1996Cell1.csv
        Source file: ../Raw Data Oct 07/Cell1/1997Cell1.csv
        Source file: ../Raw Data Oct 07/Cell1/1998Cell1.csv
        Source file: ../Raw Data Oct 07/Cell1/1999Cell1.csv
        Source file: ../Raw Data Oct 07/Cell1/2000Cell1.csv
        Source file: ../Raw Data Oct 07/Cell1/2001Cell1.csv
        Source file: ../Raw Data Oct 07/Cell1/2002Cell1.csv
        Source file: ../Raw Data Oct 07/Cell1/2003Cell1.csv
        Source file: ../Raw Data Oct 07/Cell1/2004Cell1.csv
        Source file: ../Raw Data Oct 07/Cell1/2005Cell1.csv
        Source file: ../Raw Data Oct 07/Cell1/2006Cell1.csv
        Source file: ../Raw Data Oct 07/Cell1/2007Cell1.csv
Created binary file ../Data/Cell001.mat with 395936 data records.
Processed 11 out of 1313 sensors: 0.84 percent complete.        [6.66 | 6.66]
```

[2]The complete diary file contains 5,385 lines.
[3]The MATLAB command `diary` function creates a log of keyboard input and the resulting text output.

```
>>> LINES DELETED

Processing Cell: 97
        Source file: ../Raw Data Oct 07/Cell97/1998Cell97.csv
        Source file: ../Raw Data Oct 07/Cell97/1999Cell97.csv
        Source file: ../Raw Data Oct 07/Cell97/2000Cell97.csv
        Source file: ../Raw Data Oct 07/Cell97/2001Cell97.csv
        Source file: ../Raw Data Oct 07/Cell97/2002Cell97.csv
        Source file: ../Raw Data Oct 07/Cell97/2003Cell97.csv
        Source file: ../Raw Data Oct 07/Cell97/2004Cell97.csv
        Source file: ../Raw Data Oct 07/Cell97/2005Cell97.csv
        Source file: ../Raw Data Oct 07/Cell97/2006Cell97.csv
        Source file: ../Raw Data Oct 07/Cell97/2007Cell97.csv
Created binary file ../Data/Cell097.mat with 334648 data records.
Processed 1313 out of 1313 sensors: 100.00 percent complete. [7.11 | 495.48]

Elapsed time is 505.841521 seconds.

06-Sep-2009 18:36:26: Flag_MissingData
Processing Cell 1.
Flag_MissingData: (1, 8, 534753, 1, 0)
Processed 11 out of 1313 sensors: 0.84 percent complete.         [3.56 | 3.59]

>>> LINES DELETED

Processing Cell 97.
Flag_MissingData: (112, 0, 0, 0, 0)
Processed 1313 out of 1313 sensors: 100.00 percent complete. [3.00 | 302.69]

Elapsed time is 303.535701 seconds.

06-Sep-2009 18:41:29: Flag_NotYetOperational
Processing Cell 1
Flag_NotYetOperational: (534736)
Processed 11 out of 1313 sensors: 0.84 percent complete.         [2.14 | 2.17]

>>> LINES DELETED

Processing Cell 97
Flag_NotYetOperational: (1226)
Processed 1313 out of 1313 sensors: 100.00 percent complete. [2.97 | 195.06]

Elapsed time is 194.661934 seconds.

06-Sep-2009 18:44:44: Flag_Deactivated
Processing Cell 1
Flag_Deactivated: (0)
Processed 11 out of 1313 sensors: 0.84 percent complete.         [2.14 | 2.14]

>>> LINES DELETED

Processing Cell 97
Flag_Deactivated: (69324)
```

```
Processed 1313 out of 1313 sensors: 100.00 percent complete. [2.92 | 196.06]

Elapsed time is 194.919825 seconds.

06-Sep-2009 18:47:59: Flag_TooSparseDay
Processing Cell 1
Flag_TooSparseDay: (13164)
Processed 11 out of 1313 sensors: 0.84 percent complete.         [2.36 | 2.39]

>>> LINES DELETED

Processing Cell 97
Flag_TooSparseDay: (31091)
Processed 1313 out of 1313 sensors: 100.00 percent complete. [3.20 | 217.66]

Elapsed time is 218.041631 seconds.

06-Sep-2009 18:51:37: Flag_OutOfRange
Processing Cell 1.
 1 2 3 4 5 6 7 8 9 10 11
Flag_OutOfRange: (83)
Processed 11 out of 1313 sensors: 0.84 percent complete.         [5.63 | 5.63]

>>> LINES DELETED

Processing Cell 97.
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Flag_OutOfRange: (41342)
Processed 1313 out of 1313 sensors: 100.00 percent complete. [7.16 | 508.98]

Elapsed time is 502.287523 seconds.

06-Sep-2009 18:59:59: Flag_NeighborhoodOutliers
Processing Cell 1.
        Processing Sensor 1          [39.86 | 40.36 | 40.36]
        Processing Sensor 2          [34.89 | 75.25 | 75.25]
        Processing Sensor 3          [42.09 | 117.34 | 117.34]
        Processing Sensor 4          [6.80 | 124.14 | 124.14]
        Processing Sensor 5          [7.34 | 131.52 | 131.52]
        Processing Sensor 6          [7.70 | 139.22 | 139.22]
        Processing Sensor 7          [7.59 | 146.81 | 146.81]
        Processing Sensor 8          [7.17 | 153.98 | 153.98]
        Processing Sensor 9          [6.72 | 160.73 | 160.73]
        Processing Sensor 10          [7.03 | 167.77 | 167.77]
        Processing Sensor 11          [7.28 | 175.05 | 175.05]
Flag_NeighborhoodOutliers: (948)
Processed 11 out of 1313 sensors: 0.84 percent complete. [176.89 | 176.89]

>>> LINES DELETED

Processing Cell 97.
        Processing Sensor 1          [18.88 | 19.23 | 16084.30]
        Processing Sensor 2          [23.53 | 42.77 | 16107.83]
        Processing Sensor 3          [12.80 | 55.56 | 16120.63]
```

```
        Processing Sensor 4          [23.14 | 78.70 | 16143.77]
        Processing Sensor 5          [23.19 | 101.89 | 16166.95]
        Processing Sensor 6          [13.23 | 115.13 | 16180.19]
        Processing Sensor 7          [6.80 | 121.92 | 16186.98]
        Processing Sensor 8          [8.00 | 129.92 | 16194.98]
        Processing Sensor 9          [6.70 | 136.63 | 16201.69]
        Processing Sensor 10          [7.39 | 144.02 | 16209.08]
        Processing Sensor 11          [7.69 | 151.70 | 16216.77]
        Processing Sensor 12          [14.23 | 165.94 | 16231.00]
        Processing Sensor 13          [20.83 | 186.77 | 16251.83]
        Processing Sensor 14          [8.23 | 195.00 | 16260.06]
        Processing Sensor 15          [5.56 | 200.56 | 16265.63]
        Processing Sensor 16          [8.44 | 209.00 | 16274.06]
Flag_NeighborhoodOutliers: (3094)
Processed 1313 out of 1313 sensors: 100.00 percent complete. [211.58 | 16276.64]

Elapsed time is 2565.253807 seconds.

06-Sep-2009 19:42:44: Flag_LagOneOutliers
Processing Cell 1.
 1 2 3 4 5 6 7 8 9 10 11
Flag_LagOneOutliers: (113)
Processed 11 out of 1313 sensors: 0.84 percent complete.        [221.13 | 221.13]

>>> LINES DELETED

Processing Cell 97.
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Flag_LagOneOutliers: (5833)
Processed 1313 out of 1313 sensors: 100.00 percent complete. [339.89 | 17378.47]

Elapsed time is 2701.880700 seconds.

06-Sep-2009 20:27:46: CreateSubsetRange
Computing the quartiles for each subset for every time interval.
Processing Subset (1, 1, 1):  1 2 3 4 14 15 16 17 18 19 20 21 22 23 60 62 93 95 97
Processed 49 out of 1313 sensors: 3.73 percent complete.        [22.52 | 22.95]

>>> LINES DELETED

Processing Subset (3, 13, 1):
Processing Subset (3, 13, 2):  32
Processed 1313 out of 1313 sensors: 100.00 percent complete. [3.53 | 729.05]

Created:../Data/SubsetRanges.mat
Elapsed time is 751.776146 seconds.

06-Sep-2009 20:40:18: Flag_PointExtremes
Processing Cell 1.
Flag_PointExtremes: (489)
Processed 11 out of 1313 sensors: 0.84 percent complete.        [3.45 | 9.91]

>>> LINES DELETED
```

```
Processing Cell 97.
Flag_PointExtremes: (110502)
Processed 1313 out of 1313 sensors: 100.00 percent complete. [4.47 | 312.89]

Elapsed time is 309.695131 seconds.

06-Sep-2009 20:45:28: CreateDailyRange
Processing Cell 1:  1 2 3 4 5 6 7 8 9 10 11
Processed 11 out of 1313 sensors: 0.84 percent complete.        [2.39 | 2.52]

>>> LINES DELETED

Processing Cell 97:  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Processed 1313 out of 1313 sensors: 100.00 percent complete. [5.81 | 357.91]

Created:../Data/DailyRanges.mat
Elapsed time is 367.208471 seconds.

06-Sep-2009 20:51:35: Flag_DailyRange
Computing the upper and lower bounds on the daily ranges for each subset.
Processing Cell 1
Flag_DailyRange: (2210)
Processed 11 out of 1313 sensors: 0.84 percent complete.        [4.53 | 7.86]

>>> LINES DELETED

Processing Cell 97
Flag_DailyRange: (43162)
Processed 1313 out of 1313 sensors: 100.00 percent complete. [5.47 | 417.36]

Elapsed time is 417.538977 seconds.

06-Sep-2009 20:58:33: Flag_DailyExtremes
Computing the upper and lower bounds on the daily extremes for each subset.
Processing Cell 1
Flag_DailyExtremes: (3524)
Processed 11 out of 1313 sensors: 0.84 percent complete.        [4.52 | 8.23]

>>> LINES DELETED

Processing Cell 97
Flag_DailyExtremes: (45024)
Processed 1313 out of 1313 sensors: 100.00 percent complete. [5.59 | 428.23]

Elapsed time is 428.533293 seconds.

06-Sep-2009 21:05:41: Flag_IntermittentData
Processing Cell 1.
 1 2 3 4 5 6 7 8 9 10 11
Flag_IntermittentData: (0)
Processed 11 out of 1313 sensors: 0.84 percent complete.        [2.78 | 2.78]

>>> LINES DELETED
```

```
Processing Cell 97.
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Flag_IntermittentData: (0)
Processed 1313 out of 1313 sensors: 100.00 percent complete. [4.36 | 437.20]

Elapsed time is 436.512523 seconds.

06-Sep-2009 21:12:58: Flag_InconsistentDay
Processing Cell 1.
Flag_InconsistentDay: (553392)
Processed 11 out of 1313 sensors: 0.84 percent complete.         [2.23 | 2.27]

>>> LINES DELETED

Processing Cell 97.
Flag_InconsistentDay: (1022909)
Processed 1313 out of 1313 sensors: 100.00 percent complete. [3.08 | 207.06]

Elapsed time is 207.109171 seconds.

06-Sep-2009 21:16:25: Flag_InconsistentWeek
Processing Cell 1
Flag_InconsistentWeek: (539448)
Processed 11 out of 1313 sensors: 0.84 percent complete.         [2.17 | 2.20]

>>> LINES DELETED

Processing Cell 97
Flag_InconsistentWeek: (1019910)
Processed 1313 out of 1313 sensors: 100.00 percent complete. [3.02 | 200.95]

Elapsed time is 200.874384 seconds.

06-Sep-2009 21:19:46: Flag_InconsistentMonth
Processing Cell 1
Flag_InconsistentMonth: (534747)
Processed 11 out of 1313 sensors: 0.84 percent complete.         [2.25 | 2.27]

>>> LINES DELETED

Processing Cell 97
Flag_InconsistentMonth: (1050395)
Processed 1313 out of 1313 sensors: 100.00 percent complete. [3.06 | 203.55]

Elapsed time is 203.787488 seconds.
================================================================
Elapsed time is 10510.469405 seconds.
diary
```

D.3 Execution Summary

The test platform was a Dell Studio XPS 435MT, with an Intel Core i7 CPU 920 running at 2.67 GHz, 12.0 GB of random access memory, and the Microsoft Vista Ultimate 64 operating system. The resulting computational effort, as presented in D.2, is summarized in Table D.1.

Table D.1: Required computational effort for each of the functions from the complete execution of `A_RunAll.m`, as present in Section D.2.

Function	Time (sec)	Time (min)	Percent of Total
`CreateInitialization`	0.7	0.0	0.0%
`CreateMatFiles`	505.8	8.4	4.8%
`Flag_MissingData`	303.5	5.1	2.9%
`Flag_NotYetOperational`	194.7	3.2	1.9%
`Flag_Deactivated`	194.9	3.2	1.9%
`Flag_TooSparseDay`	218.0	3.6	2.1%
`Flag_OutOfRange`	502.3	8.4	4.8%
`Flag_NeighborhoodOutliers`	2565.3	42.8	24.4%
`Flag_LagOneOutliers`	2701.9	45.0	25.7%
`CreateSubsetRanges`	751.8	12.5	7.2%
`Flag_PointExtremes`	309.7	5.2	2.9%
`CreateDailyRanges`	367.2	6.1	3.5%
`Flag_DailyRange`	417.5	7.0	4.0%
`Flag_DailyExtremes`	428.5	7.1	4.1%
`Flag_IntermittentData`	436.5	7.3	4.2%
`Flag_InconsistentDay`	207.1	3.5	2.0%
`Flag_InconsistentWeek`	200.9	3.3	1.9%
`Flag_InconsistentMonth`	203.8	3.4	1.9%
Total	10510.5	175.2	100.0%