Data Quality in Time Series Data An Experience Report

Ralf Gitzel
ABB Corporate Research
Ladenburg, Germany
ralf.gitzel@de.abb.com

Abstract

Analytics based on sensor data is of increasing interest for a variety of industries. In the context of industrial manufacturing, the goal is very often to reduce downtime through improved maintenance and to increase the output of high quality goods. However, data quality issues can lead to wrong decisions being made even when the analytics algorithm applied is correct. In this case report, an approach to data quality analysis for time series data is presented. The focus of the paper is on the best way to select and present metrics and the pitfalls which were encountered during the example project.

1 Introduction

Even before the advent of the industrial internet or "Industrie 4.0", there has been the vision to use data generated by production assets to predict failures or production problems. Modern industrial robots, motors, pumps etc. are equipped with sensors (i.e. condition monitoring systems) and in some cases connected to a distributed control system. Both systems provide data that can be analyzed for patterns that occur before actual problems manifest. The ability to predict failures or production problems enables a company to sell predictive maintenance services, to detect and eliminate design problems, or at the very least to prevent consecutive damage.

At the same time, there are increasing concerns about data quality in our company. In a typical project, the majority of time is spent on the preparation of data before the actual analyses can be started. While there are data quality tools available, there is still the need to define concrete rules that address the peculiarities of time series data from control systems and condition monitoring systems. Moreover, many problems are not easily spotted but require a detailed analysis to be discovered.

As a consequence, a software tool is required, which can detect at least the majority of data quality problems. It must be adaptable to individual problems yet leverage the commonalities of the domain of time series data. Furthermore, it must produce actionable advice that leads to genuine improvements in the data.

This paper starts with a short summary of the project background (section 2) as well as an overview of data quality problems associated with time series data which are described in the literature (section 3). In section 4, the proposed approach to the calculation and visualization of data quality metrics is described using real-world data as an illustrative example. I also explain the basic data structure used to create an extensible data quality analysis tool. The paper concludes with a discussion of the lessons learned.

2 Project Background

There is a wide consensus at our company that large parts of the preparation process required to do data analysis can be simplified and sped up through a library of automated metrics. I was given the responsibility to investigate possible designs for such a library. Development of the library has started and I have applied different versions of it to machine/asset data from various sources. In this paper, I describe my practical experiences with data quality analysis for time series data. These experiences can be of interest to researchers who plan to develop new analysis algorithms and need to understand the quality issues of their input data. The lessons learned in this paper are based on the following analyses:

Some basic lessons about the complexity of presenting data quality results where learned using data from CMMS (Computerized Maintenance Management System), warranty claim and SAP databases. The remaining data (which is the focus of this paper) was extracted from two different condition monitoring systems, looking at a total of 53 units (5 monitored by the first, 48 by the second monitoring system design).

3 Data Quality Problems and Their Origins

There is a large body of works on data quality in general (see [1], [2], [3], and [4]). The most popular approach to data quality is to use a large set of metrics which are grouped into data quality dimensions. Not all dimensions and metrics are applicable to all data sets and analytics problems.

While some data quality problems are quite obvious choices for metrics (e.g. missing data), other metrics require some understanding of the problem domain. In the literature, there are certain problems that are associated with time series data (see [5],[6],[7],[8],[9],[10],[11],[12],[13], and [14] as well as the summary shown in Figure 1). It should be noted that not all of these problems can be detected easily and there are problems which cannot be detected and/or corrected at all.

Problem Name	Problem Description	Hubauer et al.	Honeywell	Pastorello et al.	Bastos et al	Rahman et al	Guo and Liu	UNESCO	Esswein et al.	Faier and de Seixas	Colditz et al
Event Data Loss	There are gaps in the event data/time series	X	X	X	X		X	X			X
Values out of Range	The values are out of range for the domain under observation (e.g. subzero temperatures in a hot process). This can be applied to individual values but also to averages, minimums and maximums	X		X	X	X	(X)	X		(X)	
Value Spikes	Spikes or sudden changes which are implausible for the domain. (Recognize through gradients and max deviations)	X			X	X	X	X			
Wrong Timestamps	Timestamps are wrong	X	X	X			(X)	X			
Slightly Inaccurate Measurement	The value is slightly wrong which might result in the detection of a trend etc.	X	X	X				X	(X)		
Rounded Measurement Value	The value is not to the optimal level of detail or has slight variations. (This might be hidden bebind values such as 1000 or due to scale of units (e.g. m3)	X	X	X				X			
Signal Noise	Small changes which are not in the process but result from inaccurate measurements. (Recognize with low pass filter)	X		X				X	(X)		
Data Not Updated	Data is not up-to-date. (Sensors might still display old values.)	X	X				X				
Unreliable Data Source	The data source is not considered fully reliable				X	X		(X)	(X)		
Divergent Despite Correlation	Values which are normally correlated behave unexpectedly			X		X					
Units of Measurements	The units of measurement are the same for all data sources (cm vs. inch)		X					X			
Forced/Calculated Values	Compensated values are used instead of real measurements. This is only a problem if we "discover" those connections later.		X	X							
Prior Changes	Has the data been changed before?		X					X			
Data Formats	Different data formats, e.g. float vs. string etc.	X					X				
Name of Events	Different names/text for events of the same type	X					X				
Timestamps Format	There are different fomats used for timestamps which make the comparison difficult.	X									
Divergent Measurements	Values which should be the same are different. Always or sometimes.	X									
Signal Alteration	Two signals "trade places"	X									
Missing Foreign Keys	Foreign keys are missing	X									
Short Data History	The history of recorded data is too short for a good analysis	X									
Aggregated Data	Is data instantaneous or is it already an average over a certain time span		X								
Diverging Sampling	Different sampling rates in the same time series can lead to problems (e.g. how many values to put into one day?)			X							
Different Accuracy	There is a different level of accuracy for the same type of data			X							
Inconsistent Noise Level	The level of noise changes over time or from different data sources			X							
Class Imbalance	There is a bias in the sample as opposed to the population					X					
Heteroscedasticity	There are subpopulations that have different variabilities from others. (Detect via Goldfeld-Quandt test)									X	

Figure 1: Data Quality Problems in Time Series Data

There are several origins for problems in condition monitoring and other sensor data. A major problem is that sensor data is often transmitted and can be delayed or even lost due to network issues. This can lead to wrong timestamps and missing data. At the same time, the measurement results can be corrupted during transmission.

The accuracy and quality of calibration of the sensor is another important factor. In a context where minute details affect the decision whether to repair/replace equipment or not, even small inaccuracies can lead to wrong conclusions. Since in many scenarios cost is an important constraint, low-cost sensors of lesser accuracy tend to be used.

One would assume that fixing data quality issues automatically is a good approach, however, not all "fixes" really improve data quality. For example, a system might interpolate missing data to close the gaps. Depending on the analytics application and algorithm, the corrected values might avoid computational problems or might be interpreted as "interesting situations" which in turn misleads the algorithm.

Finally, when data from different sources is combined, issues such as different sampling resolutions can introduce artifacts into the data. If data has different formats, there might be conversion errors (e.g. centimeters vs. inches).

4 Measuring Data Quality

The purpose of my data quality library is to understand the data quality problems present in the data and then take measures to fix/circumvent the problems. In the worst case, data quality is so poor that using the results of the data analysis is not recommended. One of the aspects of data analytics which I consider highly problematic is the perceived quality of the results. Often, once data has been corrected to such an extent that is can be used with the analytics algorithm, the results will look "clean" and accurate. However, if the data contains a lot of interpolated and corrected data points, there is a high risk that the analysis results will strongly reflect the assumptions made and thus might no longer represent reality.

The (still ongoing) development of the library described in this paper is a highly iterative process. Over time, I have identified the following design goals which I consider necessary to run an effective data quality analysis:

- The large amount of information provided by data quality analyses has to be reduced to provide an understanding of general quality, i.e. is a dataset ready to use, worthwhile improving, or not suitable? Even a very basic analysis can result in more than 100 metrics and just providing a list is too confusing to derive immediate and effective action.
- On the other hand, individual metrics are needed to understand the nature of the data quality problems. A high-level overview indicates the areas where no attention is required but without a drill-down the real problems cannot be understood.
- Information has to be actionable. This means that there needs to be concrete information about problems, their exact location, and how to fix them.
- Metrics need to be configurable for individual data sets and analysis objectives. There is a lot of similarity in time series but there are different formats, different relevancies of fields etc.

These design goals are slightly at odds as they require increasing levels of detail. My solution was to create a hierarchy of metrics. (This is based on our work with non-time series as detailed in [15].) In the remainder of this section, I will describe the visualization concept and the architecture of the R library.

5 High-Level Visualization

In order to achieve both a good overview and the ability to drill down to the level of individual problems, a hierarchy of metrics was created. At the lowest level, individual metrics are calculated which detect the data quality problems described above and measures the overall quality as a value ranging from 0% (bad) to 100% (perfect). Due to the fact that each metric needs to describe a very precise and actionable problem, the number of metrics can easily be in the hundreds. For example, a check whether there are empty fields in the data will lead to 10 metrics for a measurement with 9 values and a timestamp. A plausibility range will add another 10 metrics and so on.

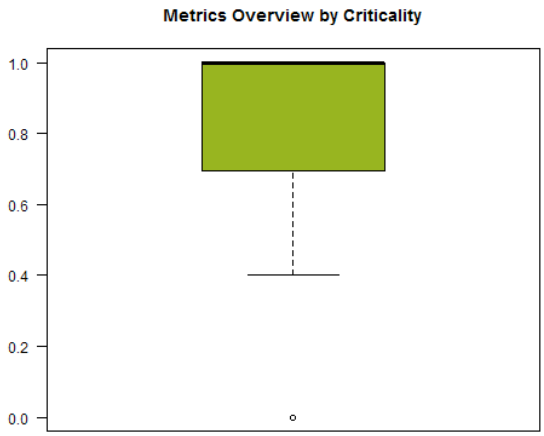

Figure 2: Overview of all Metrics

Figure 2 shows the top level overview of all data quality metrics. All metrics results (i.e. values between 0% and 100%) are shown as a boxplot. In our example, the boxplot shows that most metrics are quite OK (mean is close to 100%) but there are some poor performers and one outlier at 0%. The top level can either contain all metrics or group the metrics by importance. For example, metrics could be classified as Critical, Increased Accuracy, and Added Value with each group represented by one boxplot. In our example, all metrics had to be considered critical, thus, there is only one box-plot.

At this level, it is possible to see whether overall data quality is good or bad. In this case, it seems quite good but the outlier might require a lot of effort. By comparing several overviews from different data sets, it is possible to identify low hanging fruits for pilot tests. This is especially useful when new algorithms are being developed.

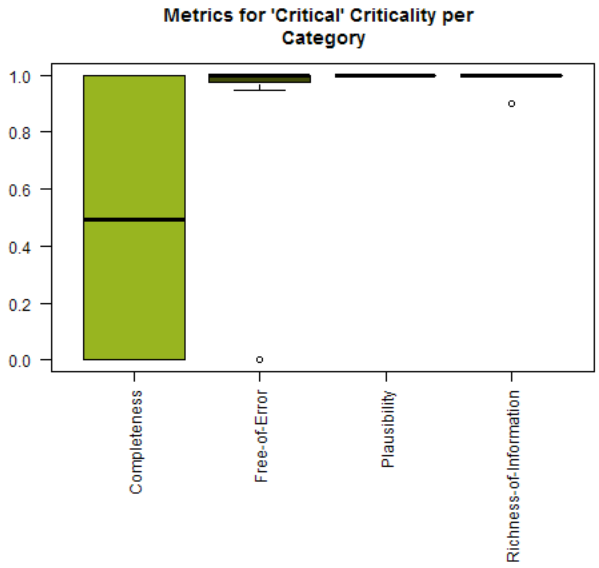

Figure 3: Data Quality Dimensions

When drilling down to the level of data quality dimensions, one gets a better understanding where the problems lie (see Figure 3). In this case, there seems to be one major problem in the category Free-of-Error and multiple problems with Completeness. There are few or no issues in the categories Plausibility and Richness-of-Information, which can thus be ignored. This level of display allows users to quickly drill down on the major problems. In this case, it makes little sense to look at any of the plausibility metrics for example.

The next level of drill-down leads to individual metrics. In the example case, there are two areas of interest. Figure 4 shows 9 of the completeness metrics. (Some identifying information has been removed for the sake of anonymization.) As can be seen, one of the major completeness problems is that data is not only missing but also that missing values are highly dependent on each other. This suggests that there might be a common cause for missing values which needs to be taken into account during the correction of the data.

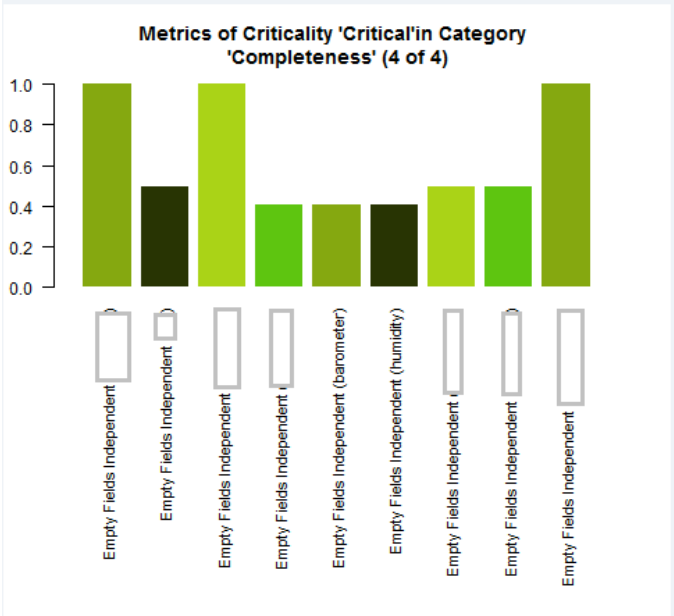

Figure 4: Completeness Metrics (Excerpt)

Furthermore, in time series, it is also interesting to see where the problems occur. For example, if all problems are concentrated in the beginning of the data, it *might* be an option cut off the initial part of the data and just use the rest for training, testing and similar activities[1]. Figure 5 shows a heatmap that illustrates the "Empty Fields Independent" metrics for all columns of the data set. The columns of the heatmap correspond to the columns of the table with data. The rows are aggregations of multiple rows in the table. So, if we go down a column we can see how the data quality of a certain value changes over time. The darker the color of the heatmap, the lower the data quality.

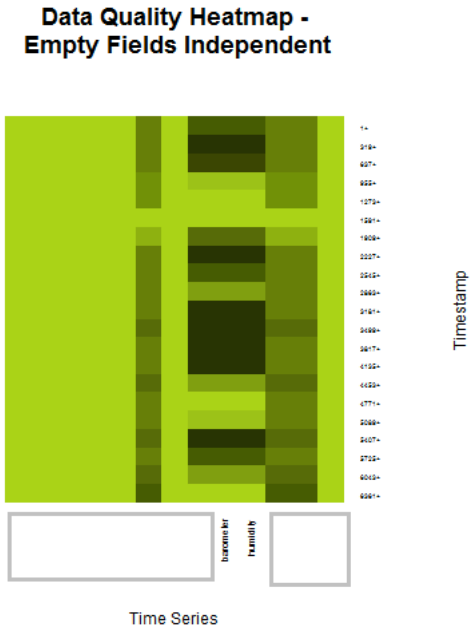

Figure 5: Heatmap of a Concrete Completeness Problem

In the example, we can see that the missing data for pressure ("barometer") and humidity are not very independent. Also, the problems are spread throughout the time series so cutting off the beginning or end is not an option.

[1] Of course, if there is some causal relationship which involves both missing data and equipment deterioration, this would be a bad idea.

6 Additional Views of the Data

If the timestamps are not equidistant, the heatmap in Figure 5 can draw a misleading picture. Let us assume for the sake of example, that a system produces 10 entries over the mission time. At first, it generated one entry per minute but the last 5 entries are at a rare of one entry per hour. In this case, the first half of the heatmap does not represent 50% of the mission time but only 10%. Thus, it is also of interest to understand the development over time, which can be vastly different from the "per entry" view.

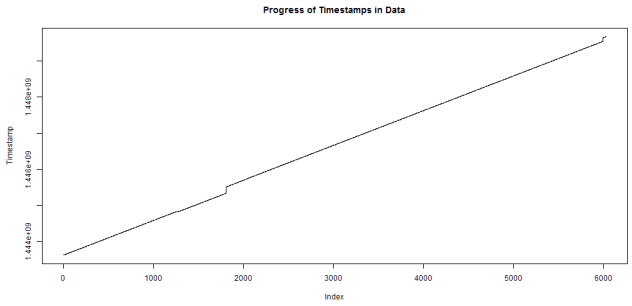

Figure 7: Progress of Time Stamps in Data

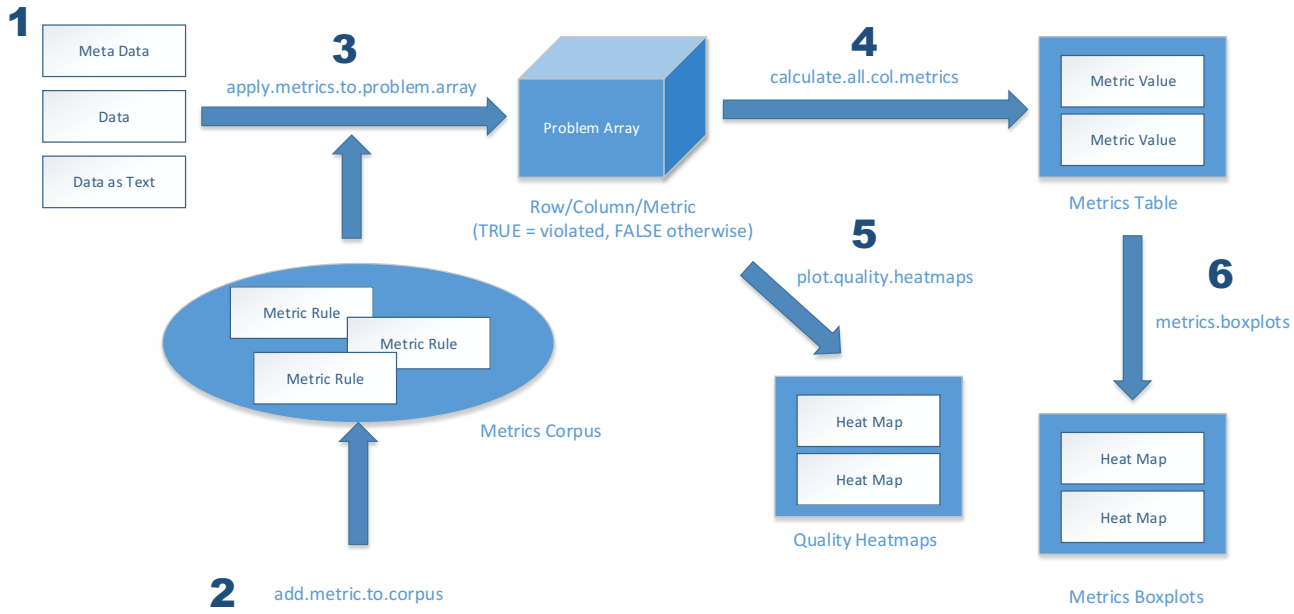

Figure 6: Overall Design Concept

Figure 7 shows the relationship between time stamps and columns. In particular, it shows how time stamp values (y axis) change as the rows of the table progress (x axis). A completely linear function means that all time stamps are equidistant and in the right order. In this case, there are two minor slips but nothing to be considered a major problem. (I.e. there are no longer periods without updates or problems in the order to timestamps.)

7 R Library

Since data sources and analysis tasks differ greatly, the assessment tool is implemented as a library that can be used to quickly implement data quality analysis tools for different use cases. The library is implemented in the functional language R which is a quite common tool for analytics. One goal of the package was to minimize the code needed to apply the metrics to an individual case. A typical application can be realized with the following steps (see Figure 6):

1. The data is loaded as text and in the correct format to allow different types of analysis. Also, metadata is needed to check for value limits, identify time stamps etc.
2. The required metrics are added to a list called the metrics corpus. The metrics are based on a series of rules which are provided in the library.
3. Via the apply.metrics.to.problem.array function, all metrics selected are automatically applied to the data and all problems found are stored in the problem array. The 3-dimensional problem array contains Boolean values to indicate whether a certain column in a certain row violates the rule for a certain metric or not.
4. The problem array can be used to automatically calculate all metrics which are stored in the metrics table.
5. The problem array can also be used to automatically create the heat maps indicating problematic areas in the data.
6. The list of metrics can be used to automatically create the hierarchy of box plots and other core graphs.

Overall, one or two pages of code can implement a data quality assessment which produces a series of PNG bitmap files to include in presentations. Examples of these files can be seen in the figures above.

8 Data Quality – Lessons Learned

The system described above is the result of multiple iterations and rounds of feedback. While some of the reasoning is already explained in the text above, I would like to briefly summarize the key findings of this project so far:

8.1 Data quality does not equal data quality

There are many generic data quality tools available on the market. However, an internal study has shown that these are mostly frameworks which need to be filled with rules. Domain-specific knowledge can greatly improve the data quality analysis.

As an illustrative example, consider vibration monitoring (see Figure 8). One has to understand the nature of vibration monitoring to know that if vibration is measured as 3 vectors of acceleration, the two horizontal vectors will show similar values, while the vertical vector will be affected by gravity and thus differ. Using this knowledge, it is possible to check the plausibility of two of the measurements and to verify the assumptions made about the sensor alignment. I.e., we might expect vectors y and x to be similar (see lower part of the figure). However, if x and z are similar, the sensors are mounted with a different orientation, i.e. our assumption about the direction of gravity is wrong.

Furthermore, some of the metrics with a statistics focus (e.g. sample size) or technical focus (sampling rate), are not implemented in generic data quality tools. While the tools might provide the capability to define such rules, we found no support in the sense that they would *suggest* such rules. To avoid misunderstandings, commercial data quality tools are not poor products but their strengths (e.g. volume and speed) were not critical for us and did not justify the price tag.

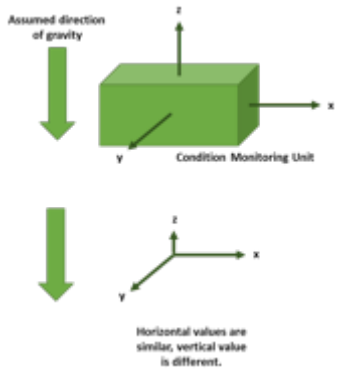

Figure 8: Plausibility of Acceleration Measurements

8.2 Understand the analytics

The importance of individual data fields depends on the type of analytics which is to be performed later. A possible outcome of training a decision tree might be that “barometer” is a very important value and “humidity” is mostly irrelevant. Fixing data quality issues is time-consuming and thus focus should be on those fields where actual value is being generated.

8.3 Impact matters

It is easy to come up with dozens of metrics. A naïve approach would be to sort all those metrics by value which quickly highlights the areas where the quality is really bad. However, it is important that the nice-to-have quality features can be separated from those of critical importance. Otherwise, mostly irrelevant poor performers will cloak essential columns which perform better but not good. Thus, the design of a good GUI is critical.

For example, let us assume a case where a large number of *richness of information* metrics is implemented. Information is rich if it contains enough detail – a statement "hot" has less richness than a statement "around 50°C" which has less richness than 49.3°C. If the richness of information is poor, this does not act as a showstopper but many metrics of this type might hide more important ones if just a list of metrics is provided. Another example is metrics which are related. Figure 4 shows that there is a strong relationship between empty fields. These metrics as well of those metrics looking at whether fields are empty or not, cover different nuances of the same problem. In a raw sorted list, such clusters of metrics will clutter up the top 10 if they are poor and might hide other important aspects.

8.4 Actionable information

During the project, one issue that came up at multiple stages was that of a "reporting trap". Interesting information was presented in graphs but when it came to the question how to fix these problems, I found that this particular information was missing or difficult to extract. Thus, besides all high-level visualization there has to be some machine-readable list which describes all problems and can thus be used to select and change subsets of the data manually or (semi-)automatically.

For example, consider completeness metrics which track the percentage of empty fields in a particular column. My initial assumption was this metric would support our decision whether to use a certain data set or not. However, reality is a bit more complex than that. For most analyses, it is not really critical to have all the values. For example, to determine a trend, a few missing values in between are not a problem. However, larger stretches without data result in unusable parts. Thus, missing 50% of the values is a problem if this means two large empty stretches. Depending on the analytics algorithm, it might be less of a problem if every other value is missing. Clearly, the metric does not give enough information for the next action – should I use this data set or a part of it or nothing at all? Which parts should I remove and which ones should I use? Only the addition of heatmaps such as the one in Figure 5 helped me to derive concrete actions from the data quality assessment.

8.5 The devil is in the details

Coming up with a good data quality analysis is no easy task. A major problem are tiny details that can cause information loss. One very interesting case was time series data where I made an assumption about time zone, only to run into problems with daylight savings time. (There were date-time combinations which do not exist and thus translated as invalid.) Also, it is easy to define the algorithms for some checks (temperature must be within a realistic range) but difficult to come up with the concrete values, even in expert interviews. (The melting point of copper is a correct but not necessarily useful upper bound for the temperature of a wire, for example.)

8.6 Some problems slip by the metrics

Even amongst those problems detectable by the metrics, there are issues which can slip by the analyst. For example, I implemented a check to discover ordering problems within the timestamps. A simple rule is to check whether timestamp n chronologically occurs after n-1. However, with this text, the following constellation (Figure 9) registers as a minimal problem because 99% of the timestamps are in correct order when compared to their neighbors. However, as the graph shows, different sections of the data are completely mixed up. In this case, some more fine-tuning is required to let the algorithm detect what a human can easily spot.

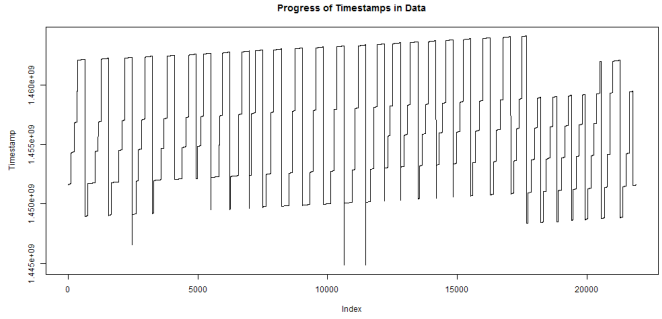

Figure 9: Undetected Major Mix-up in Time Stamps

9 Conclusions

This case study describes the development of a data quality analysis library with a focus on design issues which go beyond a description of individual metrics. In particular it addresses the aspect of quickly understanding the relevant problems and deriving concrete actions. The results shown here are but an intermediate step and hopefully the basis for a comprehensive library to be used for analytics projects in the company.

Acknowledgements

I would like to thank my colleagues at ABB for their valuable feedback. In particular I would like to express my gratitude towards Sylvia Maczey, Subanatarajan Subbiah, Michal Orkisz, James Ottewill, Ulf Ahrend, and Andrew Cohen for their input.

References

[1] Kahn, Beverly K.; Strong, Diane M.; Wang, Richard Y. (2002): Information quality benchmarks: product and service performance. In *Communications of the ACM* 45 (4), pp. 184–192.

[2] Ballou, D.P. and Pazer, H.L. (1985): Modeling data and process quality in multi-input, multi-output information systems. Management Science 31, 2, (1985), 150–162.

[3] Huang, K.,Lee, Y., and Wang, R. (1999): Quality Information and Knowledge. Prentice Hall, Upper Saddle River: N.J. 1999.

[4] Redman, T.C., ed. (1996): Data Quality for the Information Age. Artech House: Boston, MA., 1996.

[5] Bastos, M.R; Martini, J.S.C; Almeida, J.R de; Viana, S. (2010): Data integration: Quality aspects. In : Transmission and Distribution Conference and Exposition: Latin America (T D-LA), 2010 IEEE/PES, pp. 411–416.

[6] Colditz, R.R; Conrad, C.; Dech, S.W (2011): Stepwise Automated Pixel-Based Generation of Time Series Using Ranked Data Quality Indicators. In Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of 4 (2), pp. 272–280.

[7] Esswein, S.; Goasguen, S.; Post, C.; Hallstrom, J.; White, D.; Eidson, G. (2012): Towards Ontology-based Data Quality Inference in Large-Scale Sensor Networks. In : Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on, pp. 898–903.

[8] Faier, J.M; Seixas, J.M de (2010): Data quality monitoring: Independent component analysis for time series. In : Signal Processing Conference, 2010 18th European, pp. 1761–1765.

[9] Hubauer, Thomas; Lamparter, Steffen; Roshchin, Mikhail; Solomakhina, Nina; Watson, Stuart (2013): Analysis of data quality issues in real-world industrial data. In : Poster Presentation at the 2013 Annual Conference of the Prognostics and Health Management Society.

[10] Jianwen Guo; Feng Liu (2015): Automatic Data Quality Control of Observations in Wireless Sensor Network. In Geoscience and Remote Sensing Letters, IEEE 12 (4), pp. 716–720.

[11] Honeywell: Can You Trust Your Data? Understanding Data Quality is the First Step to Improving it. Honeywell. Available online at https://www.honeywellprocess.com/library/marketing/whitepapers/HoneywellProductionIntelligence_CanYouTrustYourData_WP944.pdf.

[12] Pastorello, G.; Agarwal, D.; Samak, T.; Poindexter, C.; Faybishenko, B.; Gunter, D. et al. (2014): Observational Data Patterns for Time Series Data Quality Assessment. In : e-Science (e-Science), 2014 IEEE 10th International Conference on, vol. 1, pp. 271–278.

[13] Rahman, A.; Smith, D.V; Timms, G. (2014): A Novel Machine Learning Approach Toward Quality Assessment of Sensor Data. In Sensors Journal, IEEE 14 (4), pp. 1035–1047.

[14] UNESCO: Manual of Quality Control Procedures for Validation of Oceanographic Data. Available online at unesdoc.unesco.org/images/0013/001388/138825eo.pdf.

[15] Gitzel, R.;Turrin, S., Maczey, S.; Wu, S.; Schmitz, B (2015): A data quality metrics hierarchy for reliability data. In: Proceedings of the 9th IMA International Conference on Modelling in Industrial Maintenance and Reliability (MIMAR).