MASTER

Outlier detection in event logs by using statistical methods

Lemmens, E.

Award date:
2018

TU/e Technische Universiteit
Eindhoven
University of Technology

Department of Mathematics and Computer Science
Architecture of Information Systems Research Group

Outlier Detection in Event Logs

By using Statistical Methods

Eef Lemmens

Supervisor:
Professor D. Fahland

Eindhoven, September 2018

Abstract

Process performance is a very important topic for businesses; the faster and more efficient a process runs and the less costs are associated with the process. For this reason, outliers in several characteristics in event logs give interesting results in process performance. One way to identify performance problems is through identifying outliers. Statistics provide tools to identify these outliers. In this thesis appropriate statistical methods are found to identify outliers in several aspects of the process and answer the question whether outliers in process characteristics are correlated with low performance. To find these outliers, time series created from the event logs are divided into five categories. For each of these categories an appropriate statistical outlier detection algorithm was found. Six event logs were used to validate the models. Duration of the cases that were outliers is on average much higher than for the cases that were not marked as outliers. Furthermore, we see that certain case variants are especially seen for cases that are marked as outliers. Using statistical models to find outliers in several characteristics in event logs might give interesting differences between cases that are marked as outliers and cases that are not, and thus, give a deeper understanding into where and why processes show irregularities.

Acknowledgements

First of all, I would like to thank my supervisor, Dirk Fahland, for his guidance and patience during the last months. Furthermore, I would like to thank my family and friends. My mother, my father, my brother, Inge, all my friends from the orchestra and in particular my boyfriend, Mark, without whom I would not have been able to finish my master in Business Information Systems. Their support is what kept me going. Also a big thank you to my classmates from the last two years for the group projects and studying for exams together. Lastly, I would like to thank Frans Wetzels for helping me with the secretary duties of Harmonie de Nederlanden, giving me more time to work on my studies.

Contents

List of Figures

List of Tables

1 Introduction

This master thesis tries to combine the field of Process Mining with the field of statistics to get a more in-depth insight in processes, particularly outliers that occur in several parts of the process. Process mining is an emerging field that is the link between data science and process science (Van der Aalst, 2016). Aspects of processes do not happen in a uniform way, some activities take longer than others, on some days more activities happen than on other days, and not on all days the duration of a case that starts on that day is equal. This means, there might occur outliers in the data. These outliers can be of great importance to businesses in order to make their processes more efficient, work faster, and thus, reduce costs.

1.1 Motivation

Process performance is a very important topic for businesses; the faster and more efficient a process runs and the less costs are associated with the process. Another benefit is that customers are happier when they are helped quicker. For example, if in a hospital a doctor or nurse is spending less time on administration because a more efficient way of doing this is found, he or she can spend more time with his patients and these will be helped quicker and this can, possibly, even save lives. Identifying problems in the process can give possibilities to solve the bottlenecks. One way to identify these performance problems is through identifying outliers. Statistics provide tools to identify these outliers. In this thesis appropriate statistical methods are found to identify outliers in several aspects of the process and answer the question whether outliers in process characteristics are correlated with low performance. The reason that statistical methods are chosen instead of neural networks or machine learning algorithms is that a fast approach is desirable.

1.2 Research Questions

The general objective of this thesis is to explore whether outlier analysis based on statistical techniques can give interesting insights in event logs and process performance. In order to do so, several research questions were composed. The questions will be discussed in the most logical order. First, the knowledge that needs to be present upfront will be discussed, followed by the models, and lastly the application of the models.

1.2.1 What are outliers?

This question is exploratory. But, because the term *outlier* can mean different things in different fields, it is important to discuss what is considered an outlier and what is not. If outliers are defined as values or observations that are abnormally far from the rest of the data, this gives a different starting point and result than when an outlier is defined as anything that does not seem normal. In this thesis an outlier is seen as an observation that is out of the ordinary. This research question is elaborately discussed in section 2.1.

1.2.2 In which characteristics of the event log can we determine outliers?

To get familiar with process data and event logs, and particularly the data sets at hand, we look at the different characteristics that can contain outliers. In section 2.1 characteristics such as attributes, workflow patterns, and duration are explored and discussed in detail. Some of the characteristics will be explored in a in-depth way to answer the following questions.

1.2.3 How to find outliers in the data?

Once the characteristics that can contain outliers are determined, a method to identify these outliers is needed. This research question serves that purpose. Part of the solution to the problem is given by answering this question. First, the data needs to be put in a form such that it is usable for further research. In this thesis we will mostly use time series. Subsequently, the appropriate outlier identification algorithm needs to be found. To do this, an automatic classification is made based on statistical techniques, after which the right algorithm is applied to find the outliers.

1.2.4 What is the effect of outliers on process performance?

The next question we then ask ourselves is how these outliers affect process performance. Some of the outliers will barely influence the process, while others will greatly influence it. In this thesis we will look at six different real-life event logs to see how outliers affect process performance.

1.2.5 What data gives outliers that are of importance for process performance?

This question goes even deeper than the previous question. In order to adjust the process, a company needs to know what characteristics have the biggest influence on the performance of a process. The event log of only cases that are considered to be an outlier and the event log of cases that are not considered to be an outlier are compared. These are also compared to the original log. This question is most important for business analysts. It focuses on where the process is not efficient, and thus, were businesses can safe money. In this thesis we will compare the outcome of outlier detection for several characteristics. This comparison will lead to characteristics that are more useful, and characteristics that are less useful.

1.3 Literature

This section will explain where this research fits in a broader scope. It will be compared to what other researchers already did in the field of outlier detection in statistics and how outlier detection in process data can be useful in practice. Discussing this is important for answering the question "How to find outliers?". The main focus lies on statistical methods because a fast approach is desirable.

Outliers in process performance are found for several reasons. One of these reasons is to gain insights in how the process works and were it can be improved (Berger, 2017; Teinemaa, Leontjeva, & Masing, 2015; Dees & End, 2014; Bautista, Wangikar, & Kamail Akbar, 2012). Secondly, outliers can show errors in the data. By finding outliers that are created by faults in the data, these errors can be accounted for and increase the data quality (Berger, 2017; Suchy & Suchy, 2014). This thesis tries to find outliers to improve process performance. Therefore, both reasons might be applicable for the outliers detection methods described in this thesis.

The outlier detection problem can also be reversed. It might be the case that certain characteristics of a case can cause outliers in process performance (e.g. more activities in the case, or a longer throughput time in at a certain activity) (Dees & End, 2014). If this is the case, outliers can be predicted by outlier detection methods (Chandola, Banerjee, & Kumar, 2009). Although this idea is very interesting, it falls outside the scope of this project, because we try to find outliers with an automatic method.

One kind of outliers that is widely described in research is bottlenecks in the process performance (Kooij & Rozinat, 2016; Mans et al., 2008; Van Der Aalst, 2012). Bottlenecks are the activities in processes that take relatively a lot of time. Bottlenecks are, however, not the only outliers that can be described in process performance. Other outliers are often elaborately described, but mostly they are found by simple statistics, e.g. the average throughput time of an activity, or visually (Berger, 2017; Rodrigues et al., 2017; Teinemaa et al., 2015). To deal with these outliers, they are sometimes removed for improvement of process performance (Berger, 2017). These basic statistic, e.g. averages over all cases, are considered not elaborate enough for this research. It is more interesting to look at certain characteristics on a case level; more insights in the data can be created. When only looking at the average throughput time of an activity, for example, it can only be seen which activity is a bottleneck. By looking at the average throughput time of one activity over time, we can see on which days the activity takes very long and on which days it does not. It might be the case that an activity takes on average very long, but this is caused by a few outliers of cases where this activity takes very long. By looking at the time series we can see on which days these are and even which cases they are in. By creating time series data and histograms from the process data more elaborate statistical models can be applied.

The other side of the research is statistics. A lot of outlier detection algorithms are described and applied in research (Hodge & Austin, 2004; S. Roberts & Tarassenko, 1994; S. J. Roberts, 1999; Susanti, Pratiwi, et al., 2014). Hodge et al. (2004) describes several outlier detection techniques and how they can be applied on datasets. Obviously, not all of these techniques can be applied to the data sets used for this research. First of all, Hodge et al. (2004) describes three classes of techniques; Statistical methods, Neural Networks and Machine Learning algorithms. The focus of this research lies on statistical methods, and therefore, Neural Networks and Machine Learning algorithms are not considered. The methods described in Hodge et. al (2004) are, however, not applied to process data. Chiang et al. (2002) describes several robust outlier detection algorithms that can be used on process data. The goal of this paper is to propose a new outlier detection algorithm that is computationally more efficient than other algorithms. The algorithms described by Chiang et al. (2002) are not used in this thesis because of several reasons. First of all, the algorithms described in Chiang et. al. (2002) filter out the normal operating conditions instead of the outliers in the data. The purpose of this research is to filter out outliers and see if they have any meaning within the processes. The next reason is that an outlier is defined differently than the definition that is used in this research; Chaing et. al. (2002) assumes that shutdown and start-up periods are outliers as well. The third, and most important reason is that the outlier detection algorithms described are used in on-line process monitoring techniques, which is outside the scope of this project because static datasets are used.

This thesis combines the field of process mining with statistics to find outliers automatically in process data. By finding outliers in different parts of the process data than what is done very often, more insights in the business processes can be found. This can help improve the process in a more in-depth way. By adapting the processes based on these outliers, companies can benefit from this. Businesses can get more insights in where the process shows irregularities and by trying to find out where these irregularities come from, processes can go smoother and the work can be done more efficiently.

1.4 Methodology

The methodology used for this research will be described along the CRISP DM cycle (Shearer, 2000), which can be seen in figure 1.

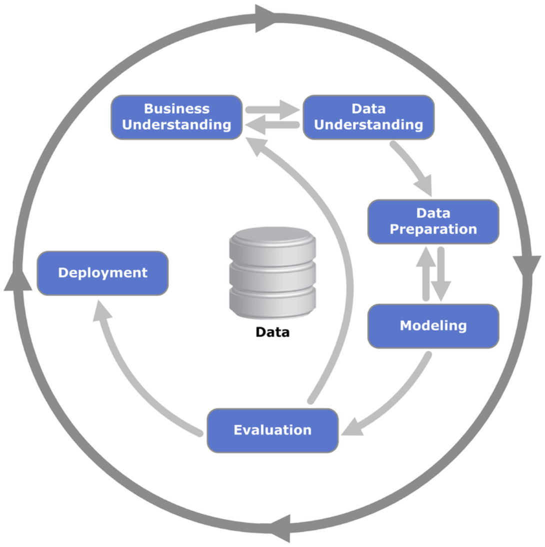


Figure 1: CRISP-DM Cycle

The first step in this circle is business understanding and data understanding. This was, of course the first step in the process that was conducted. Afterwards, data was prepared to conduct the first outlier detection algorithm, the smoothed z-score algorithm. Afterwards, the evaluation concluded that this algorithm was not suitable for every time series and therefore, more business and data understanding was needed. The analysis that was conducted resulted in five classes of time series. For each of these classes an appropriate outlier detection algorithm was found. After evaluating and optimising the algorithms that were found, the classification and outlier detection models worked fairly well and the next step was deployment. The deployment in this thesis consists of the final evaluation of the models on the data sets and insights in the process models.

1.5 Structure of this Thesis

After this introduction, this thesis consists of four chapters. In chapter 2 the preliminaries that are needed for the research conducted in this thesis are presented. Statistical methods to find outliers, time series, histograms and a discussion of outliers in event data will be discussed in this chapter. Chapter 3 discusses how time series can be categorised and chapter 4 will show why the time series are categorized in this way. Furthermore, the method to categorise the time series will be validated. In chapter 4 outlier detection techniques will be discussed for each set of time series and evaluated. Moreover, some business insights considering the outliers that were found are given. Finally, in chapter 5, a discussion and some possible future improvements are considered.

2 Background

In this chapter, the background information that is needed for understanding the rest of this thesis will be given. This includes the answer to research question 1 and part of the answer to research question 3.

2.1 Outliers

The first question that we would like to answer is the first research question: "What are outliers?". Several definitions of outliers are described in literature, a few of which will be cited here. It is important to know how an outlier is described for the purpose of this research. Hawkins (1980) described an outlier as follows:

> *"An observation that deviates so much from other observations as to arouse suspicions that is was generated by a different mechanism"*(p. 1).

According to BusinessDictionary an outlier is:

> *"Statistical data which is extremely different from the others in the same sample"*(outlier, n.d.).

The description of Greene (2012) is as follows:

> *"An observation that appears to be outside the reach of the model, perhaps because it arises from a different data generating process"* (p. 141).

In all the above definitions the word "different" is the key. An outlier is a point that is significantly different from other points in the sample. In figure 2 an example of outliers in a time series can be found. For this research an outlier is defined as *"An observation that is out of the ordinary"*. There are several reasons why outliers may occur in a time series or histogram (In section 2.2 it will be explained why only these two types of data will be considered). The first explanation would be data errors. Secondly, the data point that is an outliers could be generated by a different population and thus be generated by a different distribution than the rest of the data. This could be related to changes in behaviour of people or a system. A third explanation would be that there are unusually high residuals (Greene, 2012). Of course, outliers can also come from human error, fraudulent behaviour or instrument error (Hodge & Austin, 2004).

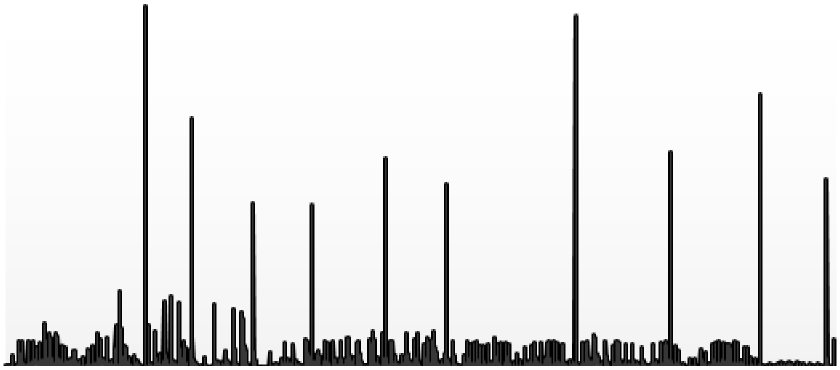


Figure 2: Outlier in a time series

2.2 Methods to find outliers

The third research question "How to find outliers in the data" will partly be discussed here. Statistical methods to find outliers in data will be elaborately discussed. This question will further be answered in section 4.2 and section 5.1. In this section the statistical methods that are available will be discussed. In section 4.2 and section 5.1 it will be explained how these statistical techniques can be applied in order to find outliers in process data.

According to Hodge et al. (2004) there are three main approaches to finding an outlier. The first, and oldest, technique is a statistical model. Several statistical models that can lead to outlier detection will be discussed. The second approach is a neural network approach and the third and last technique is a machine learning algorithm. These techniques can also be combined to reduce flaws in the methods. As the emphasis of this thesis lay on the statistical side of outlier detection, the statistical models will be described most elaborately.

2.2.1 Statistical models

The awareness of outliers started already in the 18^{th} and 19^{th} century (Harter, 1974). In 1760 Johann Heinrich Lambert presented rules for estimating the precision of a measurement both with and without the most extreme values (Sheynin, 1966). He did, however, think this was useless and never used it again. The first algorithms that were used to find outliers were statistical models that were only useful for one-dimensional data sets (Hodge & Austin, 2004). An example of a single dimensional method is Grubbs' method (Grubbs, 1969). According to Hodge et al. (2004) many statistical techniques are not able to handle an increase in dimensions in an efficient way. For this reason only single-dimensional methods are used. In some approaches a feature selection technique or a projection of the data to a subspace with less dimensions is used for analysis. Hodge et al.(2004) describe several statistical approaches to finding outliers. These approaches will be discussed next.

- **Box plot**. This technique can be used both for univariate and multivariate data sets and no assumption about the distribution is needed (Hodge & Austin, 2004). The data that is used as input for a box plot is single dimensional data. A box plot is defined by the lower extreme, the lower quartile, the median, the upper quartile and the upper extreme. The upper extreme and lower extreme are not the highest and lowest value in the dataset, but the highest and lowest value that are not considered an outlier (Laurikkala et al., 2000). According to Laurikkala et al. (2000), the upper extreme can be considered to be

$$Upper\ extreme = upper\ quartile + 1.5 \times (upper\ quartile - lower\ quartile) \tag{1}$$

and the lower extreme can then be considered to be

$$Lower\ extreme = lower\ quartile - 1.5 \times (upper\ quartile - lower\ quartile) \tag{2}$$

The output of this box plot is a graphical representation of the data in which the outliers are visually represented.

- **Proximity-based technique**. For this technique there is also no assumption about the data distribution needed. However, in contrast to the box plot, this technique becomes computationally very complex (and sometimes even unfeasible) when the number of dimensions in the data becomes bigger.
  - **Optimized *k*-nearest neighbour algorithm**. This technique produces a ranked list of potential outliers (Ramaswamy, Rastogi, & Shim, 2000). A point p is an outlier if there has less than $n-1$ other point that have a higher distance to the m^{th} neighbour than point p. For this approach, the complete distance matrix needs to be calculated. This distance can be calculated in several ways (Hodge & Austin, 2004). Two ways are the Euclidean distance and the Mahalanobis distance. The Mahalanobis distance is computationally more expensive than the Euclidean distance, because all correlations need to be calculated.
  - ***k*-means**. k clusters are formed and every cluster is represented by prototype vector that has attributes which are the mean value of all points in the cluster. The initial clustering is random according to a user-defined selection process (Hodge & Austin, 2004). After this the input data is applied in an iterative way. The algorithm looks for the cluster that matches the best and updates the prototype (and cluster centre). This minimizes the sum-of-squares clustering function.Every data point that falls outside of these clusters is considered an outlier.
  - ***k*-medoids**. This algorithm is very similar to the k-means algorithm, except it takes an actual point instead of a prototype.
- **Parametric methods**. Parametric methods are very fitting for large data sets as they are very fast in evaluating new instances. However, the downside is that a distribution model for the data needs to be selected beforehand. As for the data at hand this distribution is most of the time not known, this technique will not be used.
- **Non-parametric methods**. These approaches are more flexible and autonomous compared to parametric methods (Hodge & Austin, 2004). One example of a non-parametric method is

Binning. This consists of mapping a time series onto binary vectors by means of binning. Hence, real-valued data is transformed into discrete data (Dasgupta & Forrest, 1996). For this approach labelled data is needed. The data at hand is not labelled, therefore, this is not a suitable approach to find outliers in this data.

- **Semi-parametric methods**. These methods do not use a single distribution model like the parametric methods, but use local kernel models (Hodge & Austin, 2004). Therefore, labelled data is needed. As already mentioned before, the data used for this research is not labelled and therefore, this is not a relevant method.

Besides the approaches described in Hodge et al. (2004), there are several other approaches to find outliers. For example, a smoothed z-score algorithm, Linear Regression models, and Information Theory models (Santoyo, 2017).

- **Smoothed z-score algorithm**. This algorithm needs a time series as input. Furthermore, it needs three user defined inputs: a lag values (hereafter called lag), a threshold and an influence. The lag indicate the number of previous observations that are taken into account to smooth the data, the threshold, τ, is used to define what is an outlier and what is not, and the influence tells the influence that an outlier has on the smoothed standard deviation and average. The algorithm itself works as follows: the difference between a real data point, x, and the smoothed average from the last lag observations, μ, is calculated and compared to τ times the standard deviation from the last lag observations, σ. If the difference is higher, then the new data point is considered an outlier and it will be saved as an outlier. After, the new smoothed average and standard deviation are calculated, with the influence value if the real data point is considered an outlier (*Stack overflow Peak signal detection in realtime timeseries data*, n.d.).

- **Linear Regression model**. In order to find outliers with a linear regression model a **robust** regression model needs to be used, because other regression analysis techniques are more influenced by outliers. Ordinary Least Squares (OLS), for example, is too sensitive to outlier values (Yu & Yao, 2017). Outliers are data points with high residuals in the regression analysis. Several robust regression models have been proposed, of which the most used will be discussed here.
  - **M-estimation**. A maximum likelihood estimator is used. This method only has advantages in comparison to OLS if there are outliers in the response variable (Yu & Yao, 2017).
  - **Least Median Squares**. An estimate is found by minimizing the median of the squared residuals (Yu & Yao, 2017).
  - **S-estimation**. The S-estimate finds a plane or hyperplane that minimizes the scale of the residuals. This needs to be true for any given data point (Yu & Yao, 2017). S-estimation is more robust than M-estimation (Susanti et al., 2014).
  - **MM-estimation**. MM-estimation aims at combining the robustness of S-estimation with the efficiency of M-estimation (Susanti et al., 2014). First the regression parameter is estimated by using S-estimation. Then, an M-estimate of of the residuals based on the initial estimate is computed and the last step is to find the M-estimate, which starts at point zero (Yu & Yao, 2017).

Because these techniques are multi-dimensional, these will not be used to find outliers in event data in this thesis.

2.2.2 Neural networks

According to Hodge et al. (2004), neural network approaches are usually model-based and non-parametric. Dimensionality plays an important role in this approach as well, but it is less of a problem than in most of the statistical techniques. Neural networks focus on the key attributes by automatically reducing the number of input features. The approaches can be divided into two classes, namely supervised neural methods and unsupervised neural methods. To drive the learning process, supervised neural networks use pre-classification of the data. Unsupervised neural networks do not require this pre-classified data.

As neural networks are very difficult to train, these will not be considered in this research.

As discussed in section 1.3, a fast approach to find outliers is desirable. Therefore, single-dimensional data will be used. Next, two forms of single-dimensional data will be discussed, time series and histograms, for which we will try to find efficient outlier detection techniques in the chapters that follow.

2.3 Time series

A univariate times series shows the measurement at regular time intervals of a variable (*Online Courses 1.1 Overview of Time Series Characteristics*, n.d.). It is a sequence of measurements at successive time periods (Davidson, 2000). It is not necessary that the data needs to be independently, identically distributed for all data points, and this is thus, sometimes not the case. The ordering of the data, however, is very important. A different order of the data points could change the meaning of the data. The data should be ordered chronologically. The first data point should be the first observation in time, the second data point should be the second observation in time, etc. Time series data may have an internal structure, such as seasonality, a trend or autocorrelation. If an internal structure is present, this needs to be accounted for (*itl 6.4 Introduction to Time Series Analysis*, n.d.).

Time series data can be used for two types of analysis (*itl 6.4 Introduction to Time Series Analysis*, n.d.):

- Get an understanding of the data and the underlying forces that create the observed data.
- Forecast data by fitting a model to the observed data and continue with forecasting techniques.

In this thesis, the first type of analysis will be the reason to use time series data. The time series data will be used to get an understanding of the data and detect outliers in several aspects of processes.

2.4 Histograms

A histogram shows the probability distribution of continuous binned data (Pearson, 1895). A histogram only relates to one variable. Constructing a histogram starts with defining the intervals in which data will be divided. Afterwards, the number values that fall in the interval are counted. The intervals that are taken are most of the time not overlapping and consecutive. The data in a histogram can be normalized, such that the sum of the heights add up to 1. In this section, histograms will be further explored. First, the difference between a histogram and a bar chart will be discussed. Afterwards, more information about the properties of the x-axis and y-axis of the histogram is given.

Histogram vs. Bar chart In contrast to histograms, bar charts are based on categorical data, not continuous (or quantitative) data (Robbins, 2012). This also gives a difference in the visual representation. A histogram has bars that are adjacent, while a bar chart has gaps between the bars. A histogram can have gaps between the bars, but only when the value for a certain bar is 0. The second difference that can be seen is that a bar chart compares two variables, while a histogram shows the distribution of one variable.

Properties of a histogram A histogram can be defined by a function that counts the number of values that fall within a certain interval, or bin. This function needs to meet the condition:

$$n = \sum_{i=1}^{k} m_i \tag{3}$$

where n is the number of observations, k is number of bins and m_i is the function of the histogram. This implies that the intervals are complete. They cover the complete data set.

Binning of the data can in some sense be described as making the data categorical. The first to create guidelines for the number of bins was Sturges (1926). He defined the optimal number of bins as

$$C = \frac{R}{1 + 3.322 \log N} \tag{4}$$

where C is the number of bins, R is the range of the data (e.g. if the data ranges from 30 to 50, $R = 20$) and N is the number of observations. This formula decides automatically also on the size of the intervals.

A histogram can be made of two types of data. Namely, data that is already in the data set, and data that needs to be derived from data in the data set. If data is already in the data set, this only has to be binned to find a suitable histogram. If the data is not yet in the data set, but can be derived from it, the process takes some more effort.

2.5 Process Mining

According to Van der Aalst (2016) process mining can be described as the missing link between data science and process science. The goal of process mining is extract process related information by using event data.

Types of process mining

According to Van der Aalst (2016) there are three different types of process mining:

- *Process discovery*: A discovery technique produces a model from the event log without any a-priori information.

- *Conformance Checking*: An existing process is compared to an event log of the same process. Conformance checking is used to see if the model corresponds to reality and vice versa.

- *Enhancement*: Extending or improving a model with information from an events.

Event logs

Event logs consist of event data recorded by information systems (Van der Aalst, 2016). An event log refers to each event with a case, an activity, and a timestamp, but many other attributes may be recorded, such as resource, age, costs, etc. Event data might be stored in an unstructured form. In this case some efforts are needed to extract the event log from this data (Van der Aalst, 2016). The most used format for event logs is XES (eXtensible Event Stream), which is supported by tools like Disco and ProM.

3 Concepts, Processes and Data

3.1 Concepts

In this section several concepts that are important for this chapter will be discussed.

Van der Aalst et al. (2011) states that a **business process** "Consists of a set of activities that is performed in an organisational and technical environment. These activities are coordinated to jointly realise a business goal. Each business process is enacted by a single organisation, but it may interact with business processes performed by other organisations." The activities in a business process are performed in a particular order. An instance of a process is called a **case** (ter Hofstede, van der Aalst, Adams, & Russell, 2009). A Case consists of one or multiple **events**. An event is an occurrence of a certain **activity**. Cases are recorded by an information system that logs these in an event log. A more elaborate explanation of an event log was given in section 2.5. A case follows a certain sequence of events, which is called its case variant. A **case variant** is a specific sequence of activities (Rozinat, n.d.).

In several layers of a process in which outliers could be detected. First of all, on a **case level** outliers might be found in, for example, the duration of a case. The second level of a process is the **event level**, in which outliers could be found in characteristics such as the number of events per day. The third, and last, level that is considered is the **workflow level**. This level can contain outliers in characteristics such as patterns that are followed and number of times a loop is performed. In section a more elaborate explanation will be given in the following section.

3.2 Processes

In this section research question 2 will be answered, "In which characteristics of the event log can we determine outliers?". This question will be looked at from the data perspective in processes.

As discussed in section 3.1 there are three layers in a process that can be looked at. First of all, there is a case level, which defines the duration of a case and several attributes that belong to the case (e.g. case ID number or location). These attributes are called aggregate attributes. Furthermore, cases consist of events. The event level of a process describes several attributes (recorded attributes), the duration of the event, and of course the activity name. The attributes that are described can vary per event, but examples of these are a resource ID and amount that is paid in this event. The third level is the workflow level. A workflow does not have to consist of just subsequent activities, but can also show several patterns (e.g. a loop, an XOR pattern or a AND pattern). In all three of these levels possible outliers will be described.

Case level

First, the subjects that can contain outliers in the case level will be described. Every case contains attributes. Some attributes are always defined (e.g. duration), but some are variable (e.g. amount of a loan, age, and test results). First, the fixed attributes will be discussed and afterwards the variable attributes will be considered.

- The most obvious element in which outliers can be found is in the histogram of the duration of a case. The histogram aggregates over all cases to see if there is an outlier. If a case takes much longer or much shorter than average this is an outlier. But also in the time series of the duration of a case outliers might occur. If cases that start or end on a certain day show a much higher duration than what is considered normal, these cases can also be considered outliers.

- The next attribute that can contain outliers in some sense is the case variant. In most event logs there are several case variants. Most of the time there are a few that are very frequent, while most case variants only happen a couple of times. The variants in this last group can be seen as outliers as they deviate from most of the "points" in the sample. Also for the variants we can look at a time series. If we count the number of case variants that are done on one day, interesting outliers might show up. Of course, these two kinds of outliers can also be combined. When looking at a certain case variant the duration can be looked at.
- Other outliers can be found in the number of active cases over time or the number of activities in a case.
- In the variable attributes we can find outliers as well. Variable attributes can be divided into two categories, namely, discrete and continuous.
  - A discrete attribute can be defined by integers or categories. Examples of these are resources, or event attributes that either contain a "Yes" or a "No".
  - Examples of continuous variables are amounts of some kind. By plotting two or more of these variables against each other, or by plotting one variable against time, or by counting the number of times a certain value arises outliers might be found.

Event level

When looking at the event level, outliers can be found in the time series of several characteristics, namely:

- The number of events per day.
- The duration of events if a start and complete time are available.
- The waiting time before events.
- Variable attributes, which are obviously not the same for every event log. For these attributes the same can be said as for the attributes in the case level. Variables can be discrete or continuous and they can be plotted against each other or against time to find outliers.

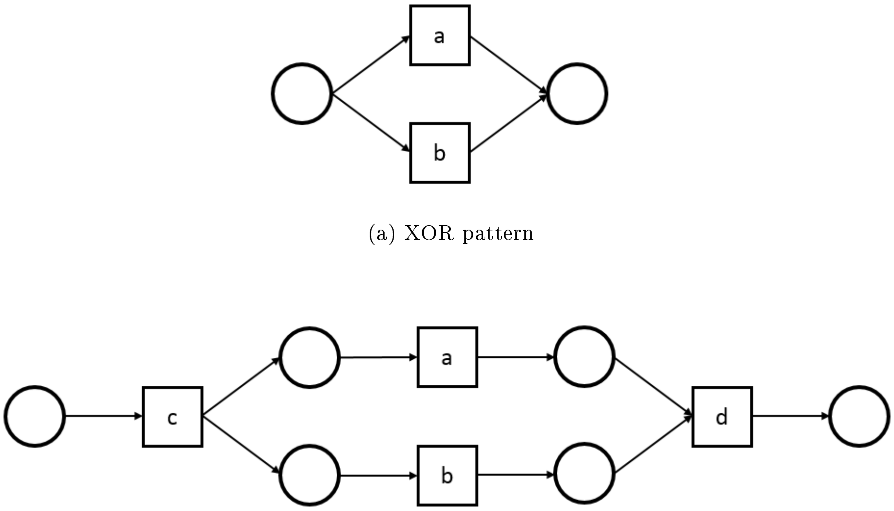
Workflow level

The third level in which outliers might be found is the workflow level. In this level we can look at

- Patterns. For example, if almost every case follows a certain pattern (e.g the pattern in figure 3a) and some cases do not follow this patterns, but another pattern (e.g the pattern in figure 3b), this last group can be seen as an outlier.

- Subsequent events. For example, when a certain activity (activity **a**) is most of the time followed by another activity (activity **b**), and in some cases this is not the case, these cases can be looked at as outliers.
- Relations between activities. For example, after deriving the directly follows graph for a process, compare the event log to this directly follows graph and cases that have more than a certain number of deviations from this might be outliers.
- The mean (or median) time of an activity in the workflow over all cases. Some activities may take on average much longer than other activities. This can, in some sense, also be considered an outlier.

None of the features from the workflow level will be used for the evaluation of the methods described in this thesis.



(a) XOR pattern

(b) AND pattern

Figure 3: Patterns

There are two types of data that can be used to make time series and histograms. The first type is recorded features. Recorded features are attributes that are already in the event log. The second type of data is derived features. These are calculated by, for example, aggregation. There are some of such time series and histograms that seem logical to derive, such as, the duration of a case and the number of active case at some point in time. However, for example, the number of times an activity within a loop is performed over time, is less obvious (see figure 4 for an example of a loop). This might be only once on average, but it might also be performed over 100 times for a case, which then constitutes an outlier. . It might also be the case that an outlier is only of significant importance if something else is the case. For example, if there is a process in which an activity is done once per month in a batch, then an outlier in number of events per day is only of importance when this is not on the day the activity that is batched happens. These derived features might give very interesting insights in the process and statistical models are able to work with the time series that are created from these. For example, cases that start on certain dates might take much more time to finish than cases that start on any other day. A company can then investigate if this has a specific cause (maybe a certain resource always works on those days, or a lot of cases start on those days causing a congestion).

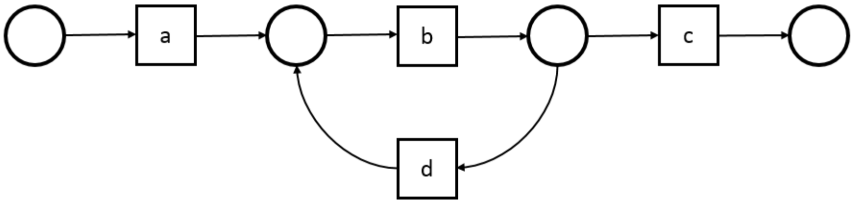


Figure 4: Example of a loop

In chapter 4 it will be explained how to derive time series from the data.

3.3 Data

Most statistical methods work best with single dimensional data. In methods that do use a multi dimensional space, data is often projected onto a subspace that has less dimensions to perform the analysis. Therefore, we chose to only analyse single dimensional data. Two frequently used types of one dimensional data are histograms and time series.

There are a few questions that can only be answered by looking at the histogram. For example, "Are there outliers in the number of events per case?", "Are there outliers in the amount that has to be paid?", and "Are there outliers in the case duration?". Of course, these last two questions can also be looked at over time. Then, the data is a time series again. However, histograms look for global outliers, while time series focus on where the outlier is exactly. To find outliers in a histogram, the distribution that it follows needs to be known (*Pqsystems Histogram: Study the shape*, n.d.). However, this distribution is not known for most of the histograms that can be created. If we look at the histogram that follows from the question "What is the time between the starting points of cases?" we can assume the arrival of cases follows a Poisson Process. Therefore, the histogram of the times between the starting points follows an exponential distribution. For the rest of the histograms, this is more difficult or even impossible to determine the underlying distribution. Therefore, we chose to focus only on time series data.

There are three layers in a process in which outliers can be detected. The case level, the event level and the workflow level. To find the outliers, we chose only to use single dimensional data, and in particular, time series. For some characteristics in these, in the next two chapters we will show how to find the outliers. In chapter 4 we will give a classification of time series and in chapter 5 we will give an appropriate outlier detection algorithm for each of these classes.

4 Time Series Classification

In Chapter 3, we derive several characteristics for which the data consists of time series. To find outliers, we first need to transform the data in such a way that it is usable for the outlier detection algorithm of choice (section 4.1). Most of the data considered here can be transformed into a time series. There are some characteristics that can contain outliers that would be histograms. In section 3.3 we explained why histograms are not used to find outliers in the data. To use the outlier detection algorithms, we needed to divide the time series into several categories (section 4.2). This is necessary to decide which algorithm can be used to find outliers in the data.

4.1 Obtaining Time Series Data from Event Data

For some of the characteristics described in section 3.2 the data can be retrieved directly from the data sets. No substantial transformation to the data has to be made. For example, for the time series in figure 5, the average number of events for cases that start on a certain day are calculated and this is plotted to create the time series. For other questions, however, a more elaborate transformation is needed. An example is the calculation of the case duration over time (figure 6). First, a set needs to be made with all dates that are in the data set. After, a start and end date needs to be found for every case and the difference between these dates needs to be calculated. Some cases span more than one day. Therefore a choice has to be made for which days those cases will count (e.g. the starting date of the case or having a particular activity on a certain day as in figure 6, where the activity *LacticAcid* is chosen as that activity). The last step is to calculate the averages for all days of the durations of cases on this day and plot these in a time series.

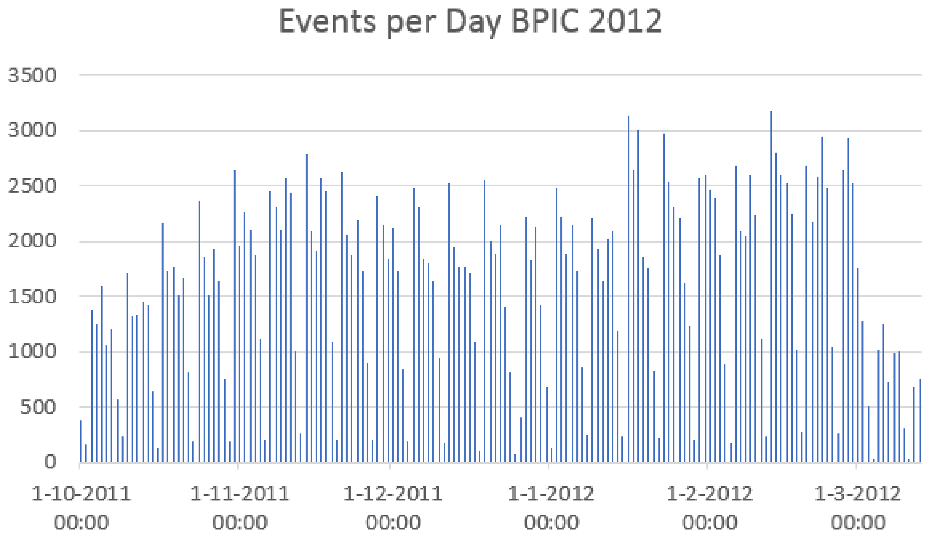


Figure 5: Example no substantial transformation

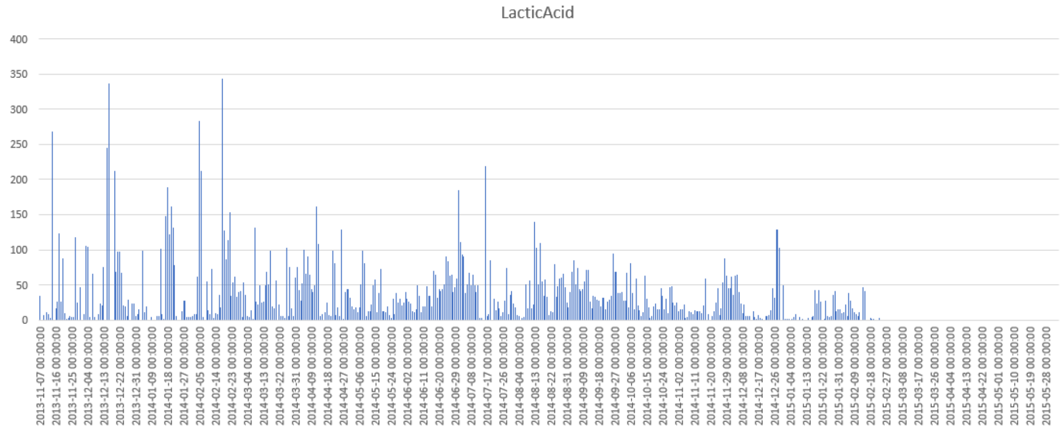


Figure 6: Example more elaborate transformation

4.2 Classification

Because the main goal of classifying the time series is to find outliers in the data, the classification is based on this idea. The following classification is chosen.

- There are some time series that have a seasonal pattern. In figure 7a an example is given of a time series that has a weekly pattern. By making these into a separate class, we can ignore the the weekly peaks (both high and low) as outliers. For example all Sundays are lower than all Wednesdays. Instead, an outlier detection algorithm can be performed on a time series that consist of the data of all Sundays. Once it is clear that a time series is in this class, we can transform the data into several new time series and these new time series can then be classified again. A monthly pattern was considered as well, but most of the datasets consisted of only several years of data. When the new time series were then created, these consisted of very little data. A more clear explanation will be given in 5.1.

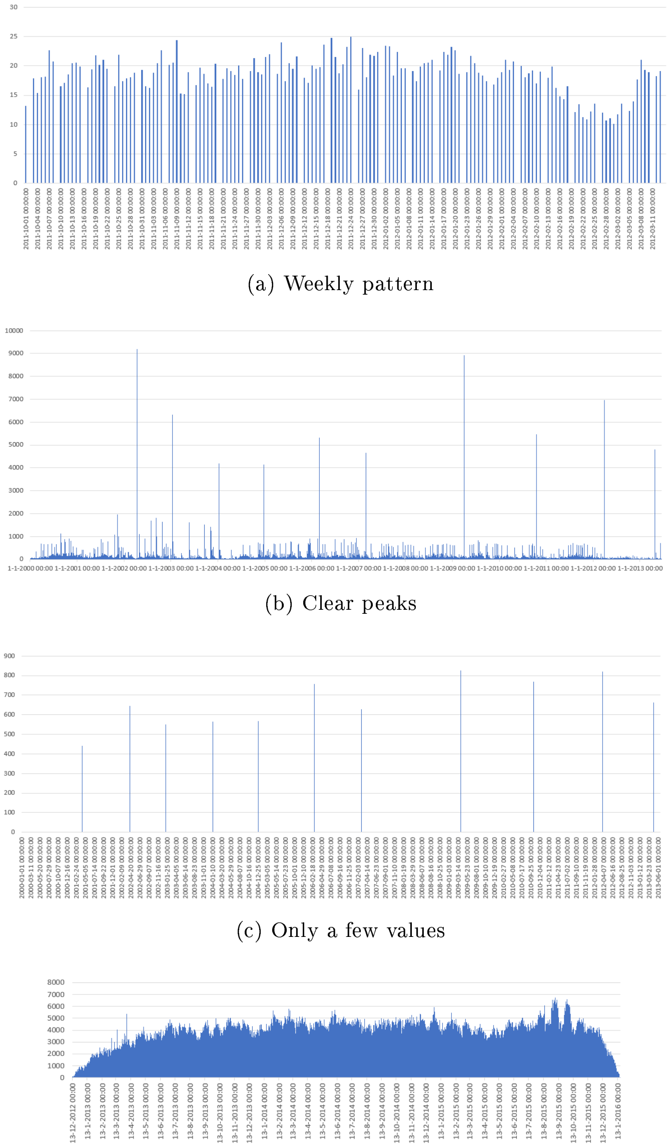
- The next class of time series has a few clear peaks (figure 7b). An outlier detection algorithm can be used for this type of time series is the smoothed z-score algorithm which will be explained in section 5.1.

- The third class that can be distinguished is time series that have only a few values (figure 7c). The outlier detection algorithm that is suitable for time series from this class will be explained in section 5.1.

- Next there are some time series that have a smooth appearance (figure 7d). The algorithm to find outliers for this class of time series can also be found in section 5.1.

- The last class are the time series that do not fall into any of the classes described before. For the datasets that are used for the validation and case study of the methods discussed in this thesis, this class only occurs for the sub time series of the time series that had a weekly pattern. The outlier detection algorithm that is applied to these type of time series is the smoothed z-score algorithm as well.

The classification is chosen in a way that the classes of time series are not mutually exclusive; a time series can be in two classes. However, the only possible combination in this is the combination of a weekly pattern and clear patterns.



(a) Weekly pattern

(b) Clear peaks

(c) Only a few values

(d) Smooth

Figure 7: Categories

4.3 Implementation

The model to divide the time series into the several categories and the models to create the time series were programmed in Python 3.

Making the time series

For every characteristic that was transformed into a time series, we needed a separate program. However, certain steps came back in every of these programs. First, we needed to create an empty time series that runs from the moment of the first event in the original event log until the moment of the last event in the original log. Next a data transformation needed to be done and with this data the time series is created.

Dividing into classes

For every category it was checked to see if the time series fitted in that category. A few remarks with respect to this will be discussed here.

First of all, to see if there is a weekly pattern within the time series, the average on every weekday is calculated. This means there is an average for Monday, an average for Tuesday and so on. These averages are then compared to each other and if there is a big difference between certain weekdays, we can conclude there is a weekly pattern.

The second class, time series with a few clear peaks, can be discovered by using an adjusted version of the smoothed z-score algorithm, discussed in section 2.2.1. The number of peaks are counted and if there are more than a certain number of peaks (this was set to 3), but less than a maximum number of peaks (30), it can be said that the time series belongs to this category. The parameters of the smoothed z-score algorithm can be adjusted and then the method discussed before will give very different results. Therefore, three different parameter settings are used:

- A lag value of 60, an influence of 0.2, and a threshold of 2.
- A lag value of 60, an influence of 0.1, and a threshold of 10.
- A lag value of 60, an influence of 0.0, and a threshold of 30.

These parameters were expert guesses.

The third class consists of the time series that have only a few values that are higher than zero. To see if a time series falls into this category, the number of values higher than zero are counted and if less than five percent of the values are higher than zero, a time series is put in this class.

The last category that we checked is the class of smooth time series. If a time series value is never more than $1,5$ times the standard deviation of the whole time series higher or lower than the previous value, a time series belongs to this category.

4.4 Validation

This section will show that the automatic classification of the histograms corresponds to what can be seen from the time series. To show the accuracy of the model, the datasets from table 1 are used.

For more information on the datasets, see appendix A. Examples from every data set will be given to validate.

Table 1: Datasets

Event Log	**Number of Cases**	**Start Date**	**End Date**
Sepsis Cases (Sepsis)	1,050	07-11-2013	05-06-2015
Hospital Log (HL)	1,143	03-01-2005	20-03-2008
BPI Challenge 2012 (BPIC 2012)	13,087	01-10-2011	14-03-2012
BPI Challenge 2017 (BPIC 2017)	31,509	01-01-2016	31-12-2017
Road Traffic Fine Management Process (RTF)	150,370	01-01-2000	18-06-2013
Hospital Billing (HB)	179,060	13-12-2012	19-01-2016

Characteristic	**Sepsis**	**HL**	**BPIC 2012**	**BPIC 2017**	**RTF**	**HB**
Events per Day	1	1	1	1	1	1
Duration of Cases	1	1	1	1	1	1
Duration of Activities	0	0	0	27	0	0
Number of Case Variants	1	1	1	1	1	1
Duration of Cases per Activity	16	0	36	0	11	18

Table 2: Time series used

There are tested 126 time series from the different data sets (see table 2) to validate the method described in section 4.2 and 4.3. The numbers in table 2 are the number of time series that resulted from the combination of characteristic and event log. The class that was automatically subscribed to the time series was compared to what was visible from the graphs from the time series. The results for all combinations of characteristics and event logs can be found in appendix B. A summary and explanation of these results will be given here. By the results from the tables in appendix B, we see that 87 time series are put into the correct category, and 39 are not. Thus, 69% of the time series are correctly classified. For some time series the automatic categorisation may find a weekly pattern that is not clearly visible from the graph. This happens for 19 time series. If we assume a misclassification in the visual inspection and ignore these time series, this would give a percentage of 81% (87 out of 107 time series) that are put in the correct category.

The main reason that time series are not put in the correct category is the settings that were chosen to categorise them. For example, it was chosen to only put time series with less than 5% of the values bigger than zero in the category of "Few Values". However, there are some time series that do have a percentage of values higher than zero that is close to 5% (e.g. 5.8%; 7.6% and 8.5%). These are now put in another category, and, therefore, outliers will be detected through a different outlier detection algorithm. Another example were settings cause an error in the category is when there are just over 30 peaks. The categorisation algorithm will most likely put this time series in the "Random" category, although it could have been classified as a "Clear Peaks" time series.

As discussed in section 4.2 it is possible that a time series belongs to multiple categories. The only combination that is possible is the combination of a weekly pattern and clear peaks in the data. These clear peaks are not always visible in the graph, or they are visible, but not important (e.g. every Sunday is seen as a downward peak in the time series). Therefore, when a time series has both a weekly pattern and peaks, but only the weekly pattern is discovered, this is not considered to be an error.

From the results shown above we can conclude that the algorithm provides us in most cases with the expected results. The cases were it did not work as expected, were mostly due to the chosen parameter settings. Therefore, we can conclude that the algorithm could be improved by optimizing parameter settings, maybe even different settings for every time series.

5 Finding Outliers

In section 4.2 we made a classification of time series. For each of these classes of time series, we need to determine an appropriate outlier detection algorithm. This will be the first subject that is discussed in this section. After this, we will give some practical examples show that these detection algorithms accurately detect outliers on real-life data.

5.1 Models

In this section the algorithms that are used to detect outliers in the time series in different categories are discussed. There are a few variables that are used in several algorithms and therefore, these presented before the discussion of the algorithms.

- f is the outlier function. The values that this function can return are 0 and 1.
- y_t is the value of the time series at time t.

5.1.1 Weekly pattern

Time series that show a weekly pattern sometimes have a few obvious outliers. For example, some processes or activities are only performed during weekdays, or during weekdays and on Saturday. On Sunday the histogram then shows that there is no activity done and the value is, thus, zero. Another example is that a certain activity is mostly done on one day of the week, e.g. Wednesday. Then, a peak is shown every Wednesday. Of course, these outliers are not very interesting to look at.

Besides the outliers that are in the data on a weekly basis, other outliers can be found. We will look at a specific weekday and see if there are outliers on that day over the weeks that data is recorded. To find these kinds of outliers, we first need to transform the data in order to be able to apply an outlier detection algorithm. When a weekly pattern was detected, seven new time series are created, one for each day of the week, and these are separately put in one of the categories from section 4.2 again. However, this new time series cannot have a weekly pattern. After this, one of the outlier detection algorithms discussed in this section can be applied.

If this same reasoning is applied to a time series with a monthly pattern, 30 new time series would be created. These would all consist of very little data for most of the datasets. For BPIC 2012, for example, this would only give 5 or 6 data points per time series. This is too little to apply a reasonable analysis to.

5.1.2 Clear peaks

For the class of time series with clear peaks the z-score algorithm described in section 2.2.1 can be used. This algorithm uses the standard deviation to find outliers in the data. The peaks in the data can either be positive or negative outliers. These outliers differ so much from the rest of the data that this algorithm can easily find them.

As discussed in section 2.2.1, the smoothed z-score algorithm contains several variables. i_t is the influence of the value at time t on the mean and standard deviation, τ is the threshold that is used, μ_t is the mean of the past lag values, so from $t - lag$ until t and σ_t is the standard deviation of the last lag values, so again, from $t - lag$ until t. The mathematics behind this algorithm is shown in equation 5 until 8.

$$i_t = \begin{cases} x & if\ f_t(y_t) = 1 \\ 1 - x & if\ f_t(y_t) = 0 \end{cases} \tag{5}$$

$$\mu_t = \sum_{k=t-lag}^{t} y_k \times i_k \tag{6}$$

$$\sigma_t = \sqrt{\frac{\sum_{k=t-lag}^{t} |y_k - \mu_t|}{lag}} \tag{7}$$

$$f_t = \begin{cases} 1 & if\ y_t - \mu_{t-1} > \tau \times \sigma_{t-1} \\ -1 & if\ \mu_{t-1} - y_t > \tau \times \sigma_{t-1} \\ 0 & otherwise \end{cases} \tag{8}$$

Several values for x and τ are used to check for outliers.

As discussed in section 4.3, we use several different parameter settings to check if a time series belongs to the category of time series with clear peaks. Once one of the parameter settings shows that a time series belongs to the category of clear peaks, we use the same parameter settings to check for outliers. So, for example, if from the adjusted smoothed z-score algorithm with parameters $lag = 60$, $x = 0,2$, and $\tau = 2$ it follows a time series falls into the category with time series with clear peaks, the same parameters are used in the smoothed z-score algorithm discussed above to detect outliers.

5.1.3 Only a few values

The time series in this class consist of only a few values. Most of the values in the time series are zero. Therefore, all values that are bigger than zero can be considered outliers. If we do so, no specific outlier detection algorithm is needed to find the outliers. When a value is above zero, it is considered an outlier. Equation 9 gives the mathematics behind this reasoning.

$$f_t(y_t) = \begin{cases} 1 & if\ y_t > 0 \\ 0 & otherwise \end{cases} \tag{9}$$

5.1.4 Smooth Time Series

For this type of time series, none of the existing algorithms were appropriate. Therefore, we created a detection algorithm that is applicable to this kind of data.

For smooth time series, we do not want to have points as outliers, but regions that are much higher or lower than average. If, for example, the z-score algorithm discussed in section 2.2.1 would be applied to the data in such a time series, this would not give any outlier points. The values are never significantly different than the average of the last 60 days (or any other number of days that is used as the lag value). For this reason, regions of high or low values are more likely to be outliers.

The algorithm that we created to find outliers in this type of time series is based on the principle discussed above. First, the overall average, the mean, is calculated by the formula given in equation

10. The mean is given by μ, and the total number of data points, y_t, is represented by N.

$$\mu = \frac{\sum_{t=0}^{N} y_t}{N} \tag{10}$$

Next, the standard deviation, given by σ is calculated (equation 11).

$$\sigma = \sqrt{\frac{\sum_{t=0}^{N} |y_t - \mu|}{N}} \tag{11}$$

After this, two values will be calculated, the mean plus standard deviation and the mean minus standard deviation (equation 12 and 13).

$$Plus = \mu + \sigma \tag{12}$$

$$Minus = \mu + \sigma \tag{13}$$

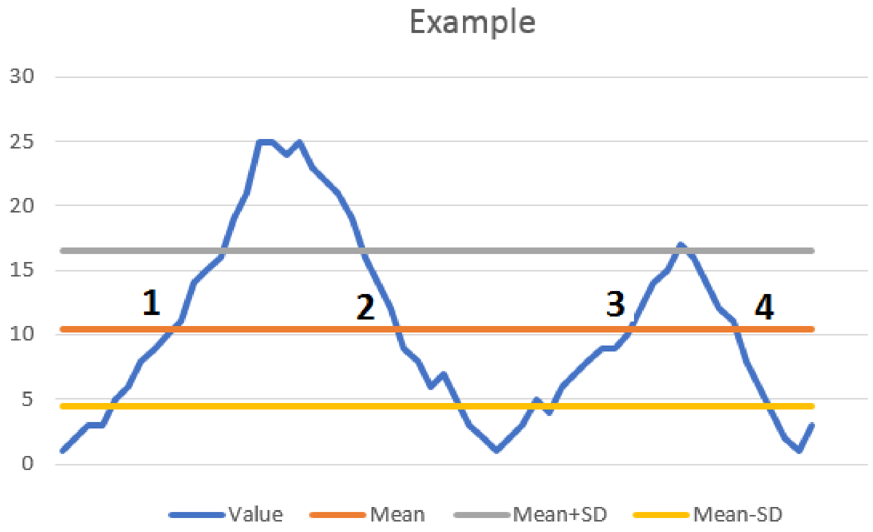


Figure 8: Example

Figure 8 gives an example of what is discussed before. The blue line is the time series at hand, the orange line is the mean, the grey line is the "plus" line and the yellow line is the "minus" line. The mean in this example is 10.48, the standard deviation is 6.00, thus, the "minus" line is at 4.49 and the plus line at 16.48.

The next step in the algorithm is to calculate the mean of periods above the orange line and periods below the orange line. In this example the mean is calculated for all points between point 1 and 2, then for all points between point 2 and 3, and finally for all points between point 3 and 4. As an example we take the period between point 1 and point 2. There are 17 values between those points and the average of these values is 18.94.

If the average of these points is higher than the plus line or lower than the minus line , all points that are higher than the plus line or lower than the minus line are considered to be outliers. However, only periods of 6 values or more are considered. So, for example, if there are only 5 data points between point 3 and point 4, this period is not considered. There can be no outliers in this period. In the example, there are more than 6 values between point 1 and point 2, and the average is higher than the grey line. So, al points between point 1 and 2 that are above the grey line are outliers.

It might be the case that a time series has several points in a row that are alternately above and below the mean (or two or three points that are above or below the mean). To make the algorithm computationally more efficient, it was chosen not to consider these short periods of time for outlier detection.

This algorithm, however, has some limitations. If there is a limited amount of data, the chosen threshold for the number of data points in a row that are above or below the mean might be too high. Furthermore, it might be the case that there is a value that is much higher or much lower than the rest. These might not be considered as outliers because the algorithm will not check single data points. An example of this is given in figure 9. There are no outliers found, while it is clear from the picture that at the start of 2017 there are some unusual values.

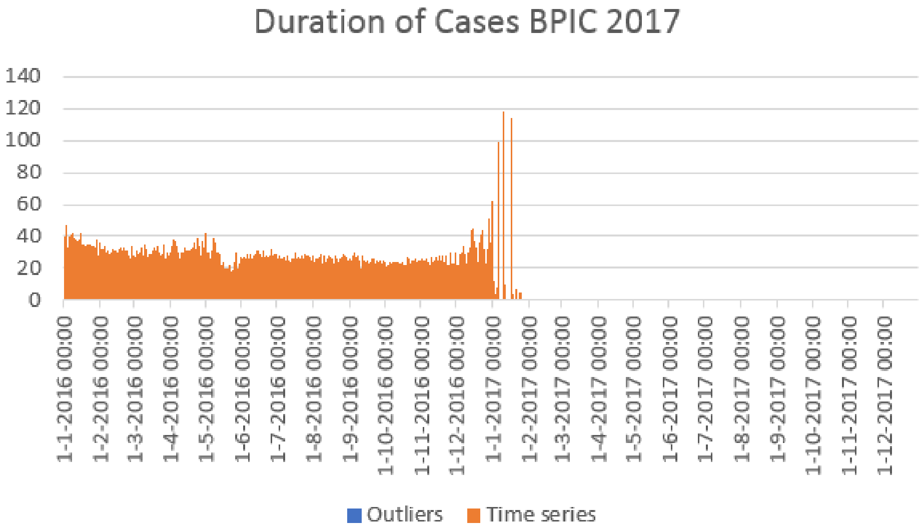


Figure 9: Example peaks in smooth time series

5.1.5 Random time series

The random time series were the time series that did not fit in any other categories. Therefore, the time series in this category are all very different. The best way to find outliers in these time series would be to apply a different, customized, outlier detection algorithm for every time series. However, the research conducted in this thesis, strives for an automatic classification of outliers. For this reason, the smoothed z-score algorithm is applied for outlier detection in this type of time series. This algorithm showed the best results. The chosen parameters were a lag value of 60, an influence of 0, 2 and a threshold of 2.

5.2 Validation

To validate the algorithms, we give one example for every algorithm in this section to show that the algorithms work. The rest of the time series that are checked for outliers can be viewed in appendix C. In figure 10 until 14 the orange values represent the time series and the blue values tell whether a value is an outlier or not. The validation is done by checking visually whether the outliers that can be seen in the time series are also classified as outliers by the methods described in section 5.1.

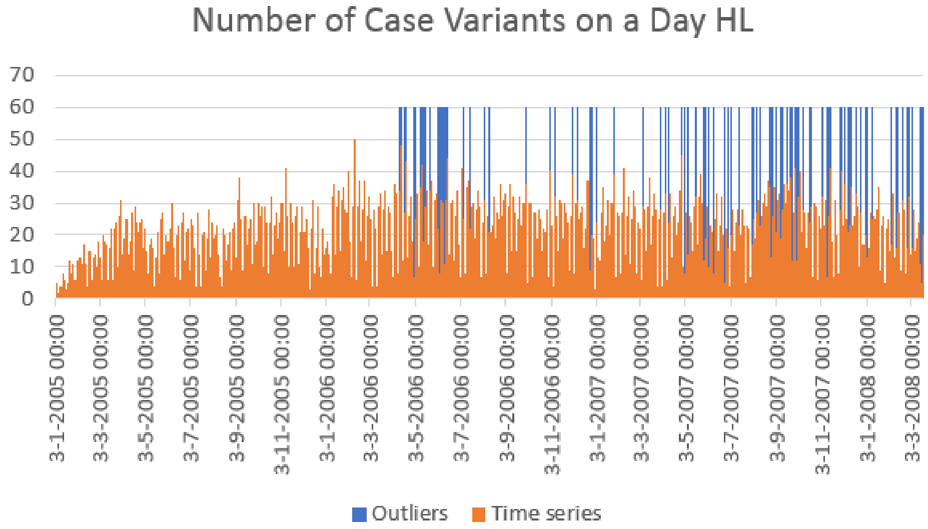


Figure 10: Weekly pattern

In figure 10, an example of a Weekly pattern is shown. This time series is split into seven new time series. For each of these time series we determine a class and afterwards the appropriate outlier detection algorithm is applied. We will validate the outlier detection algorithms that are possible next, and therefore, no validation of the outliers for this type of time series is needed.

The outliers that are found in this time series by the outlier detection algorithms are not all visually clear outliers in the time series as can be seen in the picture. We can see that there are no outliers in the first year of the period over which the number of case variants on a day are recorded. As discussed in section 5.1, a new category is found for every separate weekday. In this case, for every weekday the resulting category is either "Clear Peaks" or "Random", both of which have a start-up period of 60 days (the lag value). This explains why there are no outliers found in the first 420 days.

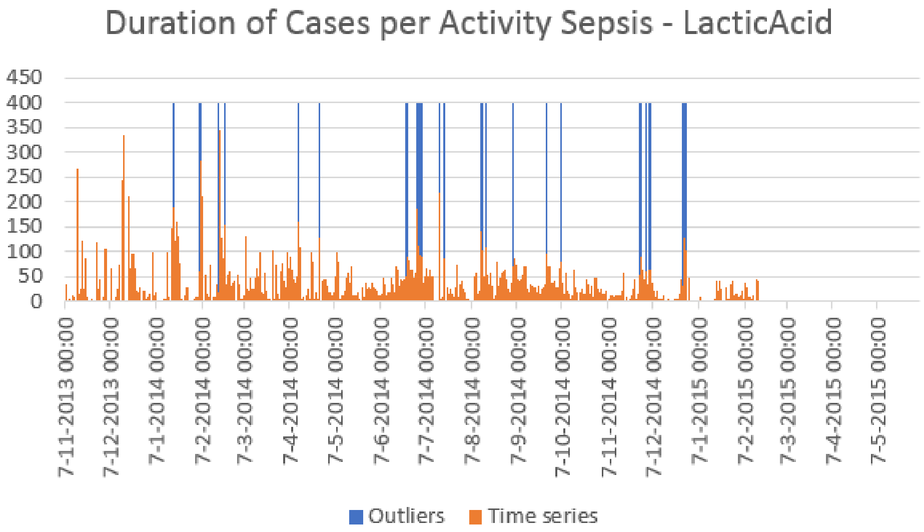


Figure 11: Clear Peaks

In figure 11, an example of the outlier detection algorithm for the next category of time series is shown; the algorithm for the clear peaks category. As already discussed in section 5.1, several different parameters are used for the smoothed z-score algorithm to find the peaks in this data. For the time series in figure 11, the smoothed z-score algorithm with a lag of 60, an influence of 0.2 and a threshold of 2 times the standard deviation is used to find the peaks. In the picture it is clear that in the first 60 days no outliers can be found. Afterwards, the local peaks are clearly found by the algorithm.

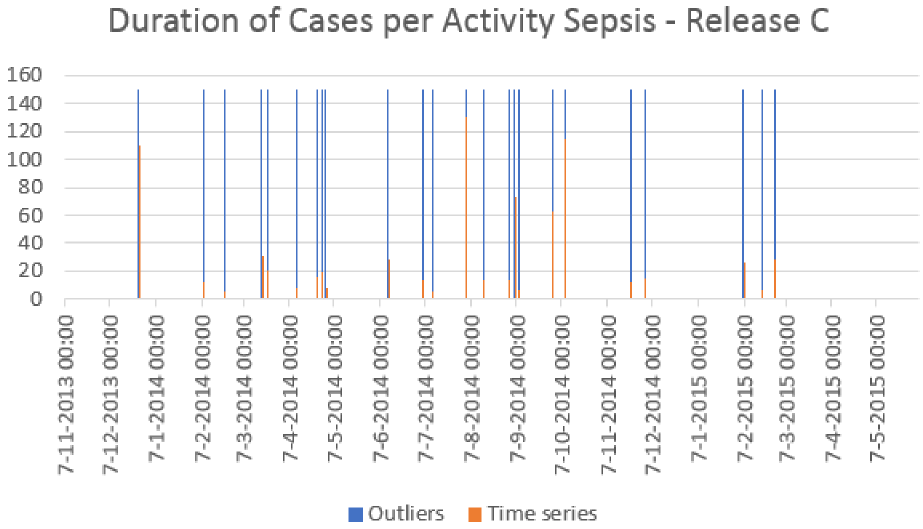


Figure 12: Few Values

An example of a time series that has few values is shown in figure 12. As can be seen, all values that are bigger than zero are marked as outliers. As there are so little values, we can say these are all outliers.

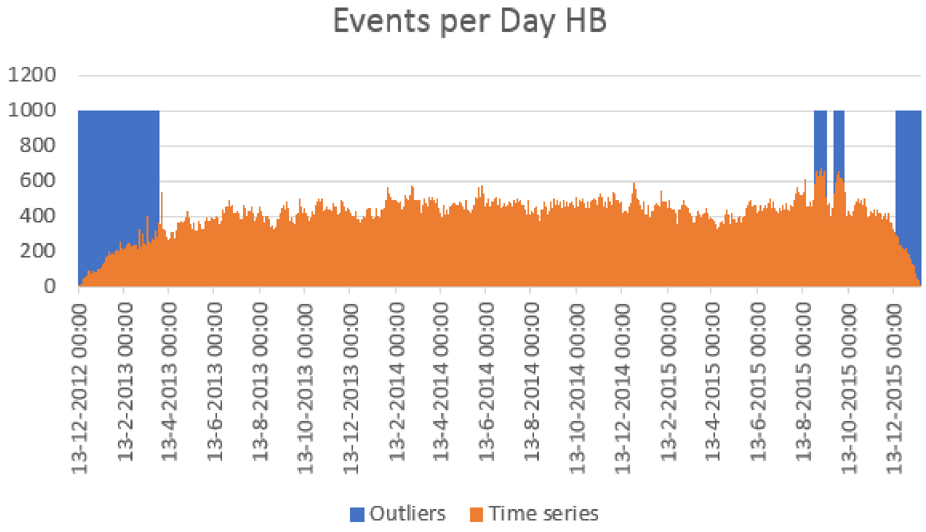


Figure 13: Smooth

In figure 13, the algorithm that is used for the smooth time series is shown. As can be seen there are regions that are marked as outliers. As expected is the start-up period marked as an outlier. There are not many values yet, and therefore, this is an outlier period with low values. The end period is also marked as an outlier period for the same reason. Furthermore, there are two regions that are significantly higher than the rest of the time series. These are marked as outliers as well. The rest of the time series looks quite smooth and around the average, which is 399. In the picture, there are other regions that seem to be outliers. However, for these regions the mean of this region is not higher than the mean plus de standard deviation and thus, this is not considered to be an outlier. Furthermore, if there are less than five values in a region, this region is not checked for outliers. The algorithm might be improved to include some of the regions that are out of the ordinary, but not high or low enough for the algorithm at hand.

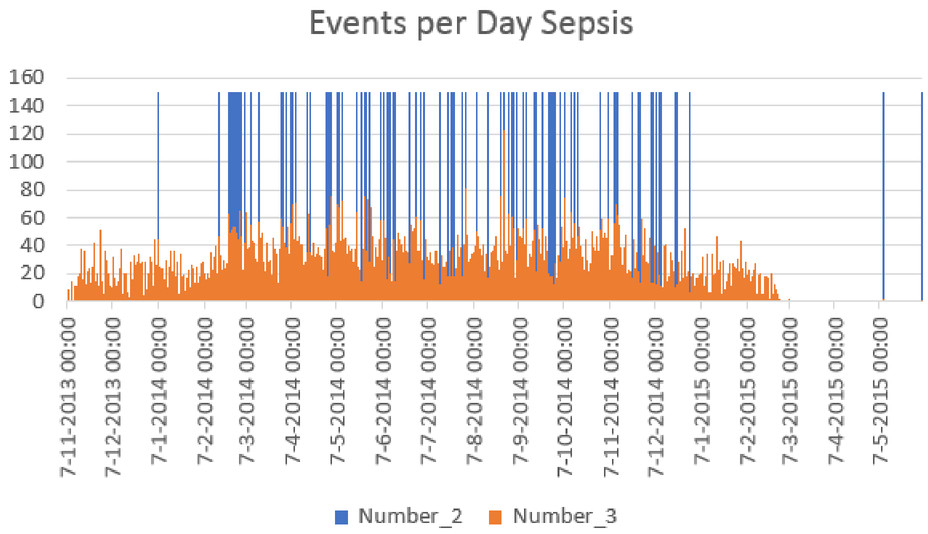


Figure 14: Random

In figure 14, an example of a time series were outliers are found with the algorithm that was chosen for the random time series is shown. The very high and very low points are marked correctly as outliers. However, there are more points marked as outliers than expected. For example between 07-02-2014 and 07-03-2014, there is a region that is completely marked as an outlier. This was not the intention of this algorithm. Therefore, improvements need to be made to find outliers correctly in this type of time series.

6 Evaluation - Meaning of outliers

In the previous chapter, we showed that the methods described in sections 4.1 and 5.1 give valid results. This section will give a more in-depth explanation of the outliers that were found. The most important question that will be answered here is "In which time series (or characteristic) can we find the most interesting outliers?" This will give more insights in how certain characteristics of processes influence the performance of the process.

6.1 Experiment setup

The experiment that we ran all needed time series data. First, we will explain how the time series that we used were constructed.

- The first type of time series, "Events per Day" is fairly straightforward. The number of events every day were counted and this was put in a time series.
- The second type of time series, "Number of Case Variants on a Day", has a similar setup. The number of different case variants that were performed every day were counted an put in a time series.
- The third type of time series was "Duration of Activities". Only the BPIC 2017 event log had start and end times for activities. Therefore, this was the only event log that was considered for this type of time series. The time series was constructed as follows. For every day, the average duration of activities that ended on a day was calculated and put in a time series.
- The last type of time series was "Duration of Cases". This time series was constructed in the following way: For every case the starting date was determined and afterwards, the average duration of all cases that started on a certain day was the value for that day in the time series.

The questions that we ask ourselves about these time series are:

1. Does the way how outliers are summarized influences the outcomes, or, in other words, impacts the share of outliers that is seen?
2. Is the mean case duration of outlier cases longer than the mean case duration of non outlier cases?
3. Do the outliers that are found have meaning in the process?

To answer these questions, various tables were made and compared. To answer question 1, we made two tables, one with the number of days that were counted as outliers for every time series and one table with the number of cases that were outliers. To decide which cases were outliers, we needed different criteria per type of time series. These tables were compared to answer the question.

- "Events per day": If a case had an event on a day that was marked as an outlier, the case was marked as an outlier as well.

- "Number of case variants": If a case had an event on a day that was marked as an outlier, the case was marked as an outlier as well.
- "Duration of Activities": We defined a case to be an outlier if and only if it had an activity that ends on a day whose "Duration of activities" values was classified as an outlier.
- "Duration of Cases": A case was an outlier if the case started on a day that had an outlier value. This was a result of how the time series was constructed.

For the second question, the original event log was split into two parts. The first part consisted of all cases that were outliers and the second part consisted of all cases that were not outliers. These two parts were event logs as well and could be compared to look not only for case duration, but also for deeper insights in the difference between the logs. The two partial event logs and the entire event log were analysed using Disco 2.1. To answer question 2, we calculated the mean duration of cases from all event logs for a every characteristic. The percentages for "Events per Day" and "Number of Case Variants on a Day" were calculated as follows:

- For every event log the average duration of the complete datafile was set as 100%.
- For every event log the duration of the outlier file and the non-outlier file were expressed in percentages of the duration of the complete event log.
- The average of all these percentages is shown in table 6.

$$Percentage = \frac{1}{6} \times Sepsis + \frac{1}{6} \times HL + \frac{1}{6} \times BPIC\,2012 + \frac{1}{6} \times BPIC\,2017 + \frac{1}{6} \times RTF + \frac{1}{6} \times HB$$

The percentages for "Duration of Cases" were constructed in the same way. However, because BPIC 2017 did not have any outliers, this dataset was not considered at all to calculate the percentages. The mean duration of the "Duration of Cases" characteristic was calculated for validation purposes. The Percentages for "Duration of Activities", naturally, only consists of the percentage of the duration of the outlier file and non-outlier file compared to the entire file of the BPIC 2017 event log.

Question 1 and 2 will be answered in section 6.2.

Question 3 links back to research question 4, "What is the effect of outliers on process performance?". The answer to this question is based on the analysis that we performed in Disco. This question will be answered in section 6.3.

6.2 Experiment

6.2.1 Does the way how outliers are summarized influences the outcomes

Tables 3 until 6 summarize some of the results that were found. A more elaborate explanation of this can be found afterwards.

Table 3: Number of outliers (days)

Characteristic	Sepsis	HL	BPIC 2012	BPIC 2017	RTF	HB
Events per Day	95 (16.52%)	130 (11.09%)	5 (3.03%)	67 (9.18%)	10 (0.20%)	166 (14.66%)
Number of Case Variants on a day	107 (18.61%)	112 (9.56%)	17 (10.30%)	56 (7.67%	1049 (21.33%)	288 (25.44%)
Duration of Activities	N.A.	N.A.	N.A.	12 (1.64%)	N.A.	N.A.
Duration of Cases	61 (10.61%)	7 (0.60%)	19 (11.52%)	0 (0%)	1004 (20.42%)	192 (16.96%)

Table 4: Number of outliers (cases)

Characteristic	Sepsis	HL	BPIC 2012	BPIC 2017	RTF	HB
Events per Day	573 (54.57%)	604 (52.84%)	546 (4.17%)	94 (0.30%)	52884 (35.17%)	39465 (22.04%)
Number of Case Variants on a day	578 (55.05%)	582 (50.92%)	1917 (14.65%)	59 (0.19%)	115326 (76.69%)	114365 (63.87%)
Duration of Activities	N.A.	N.A.	N.A.	66 (0.21%)	N.A.	N.A.
Duration of Cases	149 (14.19%)	8 (0.70%)	295 (2.25%)	0 (0%)	15881 (10.56%)	19257 (10.75%)

Table 5: Outlier detection algorithm used

Characteristic	Sepsis	HL	BPIC 2012	BPIC 2017	RTF	HB
Events per Day	Random	Weekly and Peaks	Weekly and Peaks	Weekly and Peaks	Weekly and Peaks	Smooth
Number of Case Variants on a day	Random	Weekly Pattern	Weekly and Peaks	Weekly and Peaks	Random	Random
Duration of Activities	-	-	-	Clear Peaks	-	-
Duration of Cases	Random	Weekly Pattern	Weekly Pattern	Smooth	Random	Smooth

Table 3 and 4 both show how many outliers there were found in the data. Table 3 shows the number of values in the time series that were concluded to be outliers based on the outlier detection algorithms that were described in section 5.1 and the percentage of values that is an outlier. Table 4 shows resulting number of cases that was then recognized as an outlier. Furthermore, it shows the percentage of cases that was an outlier.

From comparing table 3 and table 4 we can conclude that the percentage of values in the time series that are outliers and the percentage of cases that is an outlier can be far apart. This can happen both in the way that there are much more cases that are outliers relative to the number of values in the time series that are outliers and the other way around.

For example, the percentage of days that were outliers for the RTF event log for the characteristic "Events per Day" was only 0.20%, while the percentage of cases that were outliers was 35.17%. This implies that there are a lot of cases that have activities on the days that are considered to be outliers.

An example of a characteristic-event log combination were it is the other way around is the event log BPIC 2017 and the characteristic "Events per Day". Here the percentage of values in the time series that was an outlier (9.18%) was much higher than the percentage of cases that was an outlier (0.30%). Possible explanations for this might be that there are a lot of cases from the same case on a certain day or that the outliers are downward outliers. In the case of BPIC 2017 "Events per Day" the last one was the best explanation.

6.2.2 Mean Case Duration

Table 6: Mean Duration of Cases

Characteristic	Outliers	Non-outliers	Entire Log
Events Per Day	221.05%	80.73%	100%
Number of Case Variants on a Day	327.32%	68.98%	100%
Duration of Activities	651.99%	98.92%	100%
Duration of Cases	150.13%	95.77%	100%

Table 6 displays the mean duration of cases from all event logs for a certain characteristic. The numbers that led to this table can be found in appendix D.

From the percentages in table 6 it can be seen that cases that were outliers had on average a much higher duration than cases that were not outliers.

The biggest difference between the outlier cases and the non-outlier cases was in the "Duration of Activities". This number was, however based only one event log. When looking at the biggest difference between the outlier cases and non-outlier cases for every separate event log the biggest difference was in the characteristic "Events per Day" for the BPIC 2012 event log. The outlier cases had an average duration that was 1278.70% of the duration of the entire event log. The non-outlier cases had an average duration that was 97.83% of the duration of the entire file.

The percentages in table 6 might suggest that the duration of outlier cases was always higher than the duration of cases that were not outliers, but there were some exceptions. These exceptions only occured in "Duration of Cases". For example, the average case duration of the outlier cases from the HB event log had a duration of only 32.80% compared to the duration of the entire event log. When looking at the percentage of cases and values in the time series that were outliers (table 3 and 4), it can be seen that it was not a small number of outliers. In figure 15 the outliers that were found in this dataset can be seen. Most of the outliers that were found were in either the start-up period or in the end period. Most outliers were thus downward outliers. This caused the low percentage.

It would be interesting to calculate the statistical significance of these numbers. However, this was not possible due to the way they are calculated. If the percentage in table 6 were weighted averages, we could have calculated statistical significance. Nonetheless, we chose not to use weighted averages because of the big differences in number of cases per event log. Weighted averages would have almost eliminated some of the event logs, for example Sepsis, because this event log only has a small number of cases compared to the HB event log.

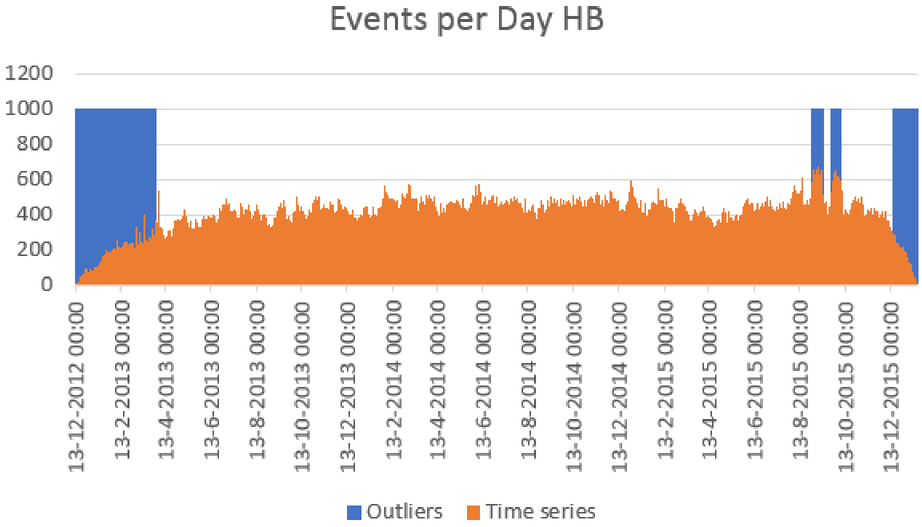


Figure 15: Events per Day - HB

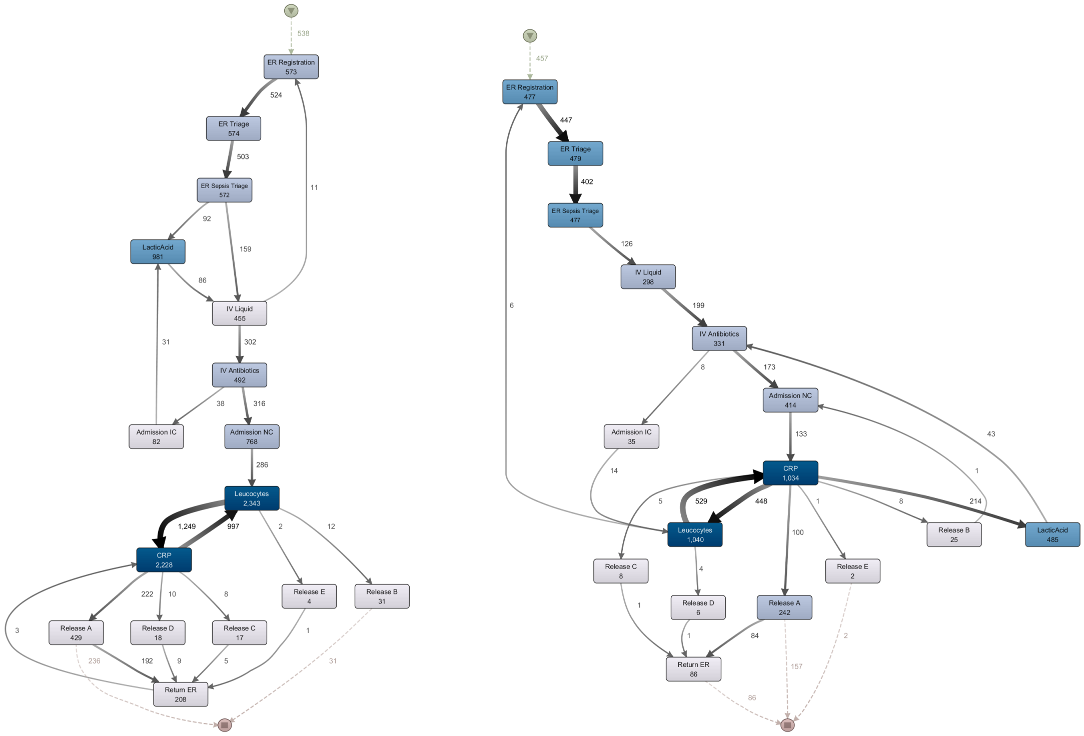
6.2.3 Correlations

It might be the case that outliers follow a pattern on themselves. However, we decided not to include autocorrelation because the time series of outliers might not be stationary. Stationarity means that the mean and variance are constant over time (Casella & Berger, 2002). For every day in a time series an outlier detection algorithm looks if that day is an outlier (value 1) or if it is not an outlier (value 0). In this way a new time series is constructed. For outlier time series that were constructed by the z-score algorithm (Clear peaks and Random), there is a start-up period of 60 days and therefore, these can not be considered to be stationary. However, after this first 60 days, the outlier time series might be stationary. This needs further investigation. The outlier time series that are constructed by the smooth algorithm have certain periods of time that are outliers and, thus, the values are one and certain periods of time that are not outliers and, thus, the values are zero. There might be a rare case that this new time series is stationary, but further investigation in these time series is needed to conclude that. The outlier time series that are constructed by the few values algorithm have the highest chance of being stationary because there are only a few values that are one. The mean in the time series will be most likely to be zero over the entire span of the time series. For the original time series that had a weekly pattern any combination of outlier detection algorithms discussed before could be used to construct the outlier time series. Therefore, there is a chance that this outlier time series is stationary, but this also needs further investigation.

6.3 Deeper insights in the event logs

In this subsection several business insights in the data will be given. Some interesting conclusions could be drawn from the numbers that were found in analysing the event logs for outlier cases and comparing these to both the non-outlier cases and the complete event logs. The insights will be given per event log. Looking at the outliers of every event log will give a more in-depth analysis of the outliers than an overall analysis.

- Sepsis: For the outlier cases, much more cases included "Admission NC" and "Admission IC" for the characteristic "Events per Day" (88.83% and 13.26% respectively). In the non-outlier event log this was only 61.0% and 7.13% respectively. In figure 16 the two process models can be seen. These activities, however, seemed necessary for patients to get better and it might not be possible to speed up the process.



(a) Outliers

(b) Non-outliers

Figure 16: "Events per Day" Sepsis Cases Process Model

- HL: Both for "Number of Case Variants on a Day" and "Events per Day" the average number of events per case was much higher for the outlier cases than for the non-outlier cases (in both characteristics almost four times as high). Next, in all characteristics a case variant was done at most three times. This means, outlier cases were correlated with many different case variant happening on the same day. For the non-outlier cases there were some case variants that were used more often. These were mostly case variants of two or three events. There were a few cases (between 30 and 40 for both "Number of Case Variants on a Day" and "Events per Day") that had more than 500 events. Therefore, to reduce costs, it might be possible to look more critical at which events (or tests in this event log) are necessary. However, since no domain knowledge is available, this might be a conclusion that is drawn to soon.

- BPIC 2012: From the outlier cases event log for the characteristic "Number of Case Variants on a Day" we could conclude that in the outlier cases more loans were approved than for the non-outlier cases (25% of the cases in the outlier cases event log against 15% in the non-outlier cases event log). In the complete event log 17% of all submitted loan applications was approved. For "Events per Day" these percentages were 26% for the outlier cases event log and 16% for the non-outlier cases event log. For the "Duration of Cases" characteristic the percentages lay closer to each other; 22% for the outlier cases event log and 17% for the non-outlier cases event log. What could be concluded was that cases that were an outlier, were more likely to be cases in which a loan was approved. For all three characteristics the duration of the outlier cases event log was approximately two times the duration of the complete event log. Therefore, the company might look for a way to speed up the process of approving a loan. However, it is understandable that this process takes longer than declining a loan. Furthermore, the work was more spread between resources for the outlier cases event logs. It might be the case that certain resources work slower than others. Waiting times between activities or the duration of activities are needed in order to see if this is indeed the case.

- BPIC 2017: At first sight, none of the findings showed interesting differences between the outlier cases event log and the non-outlier cases event log. Therefore, more research needs to be done.

- RTF: For both "Events per Day" and "Number of Case Variants on a Day" the cases in the outlier file were the cases where people did not pay immediately. Therefore, these cases took longer and had more events per case. The activity "Payment" was for the outlier cases event log for the characteristic "Events per Day" only done in 2467 out of 52884 (4.66%) cases and for the characteristic "Number of Case Variants on a Day" done in 36270 out of 115326 cases (31.45%). To put this in perspective, in the complete event log "Payment is performed in 69715 out of 150370 cases (46.36%). This means that for the characteristic "Events per Day" 50417 out of 80655 (62.51%) of the cases that did not have the activity "Payment" was in the outlier file. When looking at the characteristic "Duration of Cases" we saw that the cases in the outlier file performed "Payment" in 52.58% of the cases, while the cases that were not outliers performed this activity in only 45.63% of the cases. The average duration of the outlier cases was also shorter than the average duration of the complete event log. In figure 17 it can be seen that both the high peaks and the low peaks and several other values were seen as outliers. However, since this time series belongs to the category Random, it might be the case that the outliers were not found in an appropriate way.

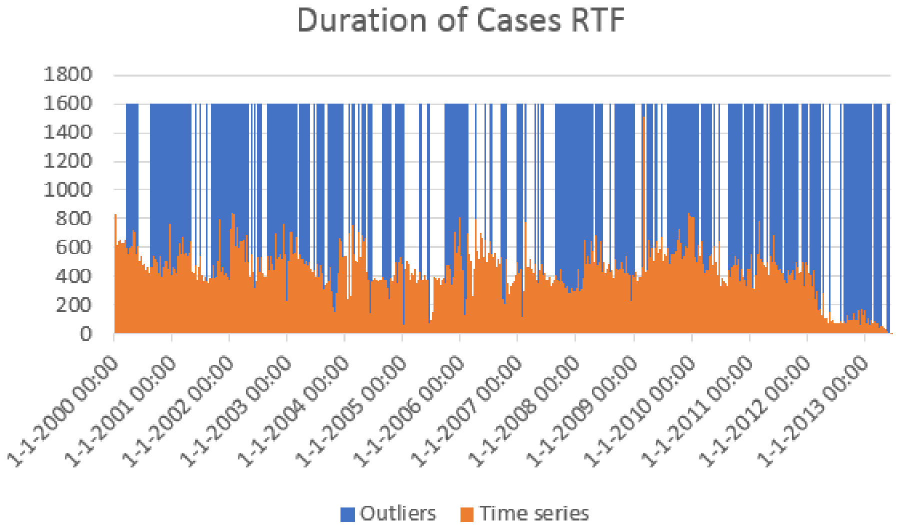


Figure 17: Duration of Cases RTF

The company could look at the cases that were outliers and see where they can speed up the process. The fastest possible route would be if people pay the fine immediately after the fine is created in the system. It would, however, be an illusion to hope that all people pay their fine immediately. However, there might either be a way to make more people pay immediately or speed up the process when people do not pay.

- HB: For the "Duration of Cases" there were less events per case on average for the outlier cases event log (only 1.94, while for the non-outlier cases event log this was 2.59 and for the complete event log there were 5.03 events per case on average). As already concluded before, the outliers found for the characteristic "Duration of Cases" were mostly downward outliers.

With the findings above, we can try to answer the last research question: "What data gives outliers that are of importance for process performance?". Both the characteristic "Number of Case Variants on a Day" and "Events per Day" gave very interesting findings for most event logs. The differences between outlier cases event logs and non-outlier cases event logs were quite significant. However, "Duration of Cases" gives for the datasets at hand far less interesting results. Both upward and downward outliers were found in the datasets, causing the results to become less coherent and more research needs to be done to explain these numbers. For "Number of Case Variants on a Day" and "Events per Day" most of the time only upward outliers were found, causing the numbers and results to be clear; mean duration was a lot higher, resource behaviour was sometimes different, and in some cases the process even looked completely different or different case variants were used. The characteristic "Duration of Activities" was only looked at for the BPIC 2017 event log, because this was the only event log that had starting and ending times for the activities. Therefore, no conclusion could be drawn to whether this characteristic would give in general interesting results or not.

Of course, there are many other characteristics that can be looked at. In section 2.1 many were discussed. Some of these, like events per case, number of times a certain attribute occurs etc., will result in histograms, and, thus, the methods presented in this thesis will not be applicable to these. To other characteristics (e.g. waiting times before next activities over time, number of times a loop is performed for a case that starts on a certain day, etc.) the models presented here can be applied.

7 Conclusion

In section 7.1 , we will first give a short summary of the answers that were found throughout the thesis and afterwards, we will give a short conclusion. The second section in this chapter will give some possible improvements on the model that was presented.

7.1 Research Questions

It is clear that these research questions cannot be answered with one simple answer or one simple algorithm. Especially questions 3, 4 and 5 do not have one simple answer.

What are outliers?

Question 1 can be answered in one sentence: *"An observation that is out of the ordinary".* This definition was used as a guideline for the rest of the research.

In which characteristics of event logs can we determine outliers?

Question 2 needed a more elaborate answer. To summarize, there are three layers in a process; the case level, the event level, and the workflow level. In the first two layers attributes can be distinguished from which time series or histograms can be made, resulting in a data type to which outlier detection algorithms can be applied. The workflow level contains patterns, subsequent events and relations between activities, which can, with some effort, also be transformed into time series and histograms.

How to find outliers in the data?

The third question could be answered with several statistical models. It was chosen to only look at time series data, because distributions for histograms were hard or impossible to find. Because all datasets are different it was not possible to choose one algorithm that could get the outliers from all datasets. Therefore, the time series were classified into five classes, and for each of these classes a suitable outlier detection algorithm was found. The classification of the time series worked correctly in 81% of the cases and the outlier detection algorithm

What is the effect of outliers on process performance?

Question 4 is answered in section 6. The main result that was found was that the duration of cases is longer for the outlier cases event logs than for the non-outlier cases event logs and the complete event logs. Furthermore, some event log specific results were given.

What data gives outliers that are of importance to process performance?

The last question was also answered in section 6. From the characteristics that were looked at "Number of Case Variants on a Day" and "Events per Day" gave the most interesting results in terms of deeper insights in the event log. However, there are more characteristics to look at, and these might even give more interesting results.

Conclusion

To conclude, using statistical models to find outliers in several characteristics in event logs might give interesting insights in the process performance. High or low outliers might be found and these can have a big influence on process performance. Comparing the cases that are outliers with the cases that are not outliers, gives a deeper understanding of were the outliers occur and how they might be solved. Business can use this to make their processes more efficient. The research questions that were asked led to a model that could automatically detect outliers in time series data that resulted from event logs.

7.2 Possible improvements

All research questions asked in section 1.2 have been answered in this thesis. However, the models presented can possibly be improved. A couple of possible improvements will be discussed in this section.

A possible improvement on the model presented in section 5.1 would be to use a different lag value for the smoothed z-score algorithm if the time series at hand has a weekly pattern. If the lag value is lower than 60, more values are considered to check for outliers. If the lag value is 60, the first 60 values will be used to set a mean and standard deviation, and are thus not checked for outliers. However, if there is a weekly pattern in a time series, seven new time series are created, which have only 1 out of 7 values from the original time series. Therefore, this might cause a problem. In the model that is presented here, if there are less than 60 values, a different lag value is used, namely, 10. But, if the new time series has 61 values, the lag value of 60 will be used. This will not give the most accurate results. For the random time series, the smoothed z-score algorithm would give more accurate results if the parameters were adjusted separately for every time series.

When the time series shows a weekly pattern, the lag value of 60 is maintained for the z-score algorithm, both for the peaks and the random time series. If every weekday shows either a peaks pattern or a random pattern, this results in the first 420 values that are not taken into account. If a dataset consists of less than two years of data, this would mean that only a few values are considered for outlier detection.

Another possible improvement might be to apply a different algorithm for the smooth time series. An algorithm that is based on the same principle as the algorithm that is described in section 5.1, but with a time window. The time window is used to calculate the mean and standard deviation and based on this, the outliers are determined in the same way as in section 5.1.

References

Baker, L. D., Hofmann, T., McCallum, A., & Yang, Y. (1999). A hierarchical probabilistic model for novelty detection in text. In *Proceedings of international conference on machine learning.*

Bautista, A. D., Wangikar, L., & Kamail Akbar, S. M. (2012). Process mining-driven optimization of a consumer loan approvals process. *BPIC 2012*.

Berger, F. (2017). Mining event log data to improve a loan application process. *BPIC 2017*.

Bishop, C. M. (1994). Novelty detection and neural network validation. *IEE Proceedings-Vision, Image and Signal processing*, *141*(4), 217–222.

Business Dictionary outlier definition. (n.d.). `http://www.businessdictionary.com/definition/outlier.html`. (Accessed: 04-04-2018)

Casella, G., & Berger, R. L. (2002). *Statistical inference* (Vol. 2). Duxbury Pacific Grove, CA.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 15.

Chiang, L. H., Pell, R. J., & Seasholtz, M. B. (2003). Exploring process data with the use of robust outlier detection algorithms. *Journal of Process Control*, *13*(5), 437–449.

Dasgupta, D., & Forrest, S. (1996). Novelty detection in time series data using ideas from immunology. In *Proceedings of the international conference on intelligent systems* (pp. 82–87).

Davidson, J. (2000). *Econometric theory.* Blackwell Publishing.

DeCoste, D., & Levine, M. B. (2000). Automated event detection in space instruments: a case study using ipex-2 data and support vector machines.

Dees, M., & End, F. v. d. (2014). A predictive model for the impact of changes on the workload of rabobank group ict's service desk and it operations. *BPIC 2014*.

der Aalst, W. M. P. V., Schonenberg, H., & Song, M. (2011). Time prediction based on process mining. *Inf. Syst.*, *36*, 450-475.

Dumas, M., La Rosa, M., Mendling, J., Reijers, H. A., et al. (2013). *Fundamentals of business process management* (Vol. 1). Springer.

Gal, A., Mandelbaum, A., Schnitzler, F., Senderovich, A., & Weidlich, M. (2017). Traveling time prediction in scheduled transportation with journey segments. *Inf. Syst.*, *64*, 266-280.

Greene, W. H. (2012). *Econometric analysis (international edition).* Pearson.

Grubbs, F. E. (1969). Procedures for detecting outlying observations in samples. *Technometrics*, *11*(1), 1–21.

Harter, W. L. (1974). The method of least squares and some alternatives: Part i. *International Statistical Review/Revue Internationale de Statistique*, 147–174.

Hawkins, D. M. (1980). *Identification of outliers* (Vol. 11). Springer.

Hodge, V., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial intelligence review*, *22*(2), 85–126.

itl 6.4 introduction to time series analysis. (n.d.). `https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc4.ht` (Accessed: 12-08-2018)

Kooij, P., & Rozinat, A. (2016). *Automation platforms and process mining: A powerful combination.* IEEE CIS Task Force on Process Mining.

Korn, F., Labrinidis, A., Kotidis, Y., Faloutsos, C., Kaplunovich, A., & Perkovic, D. (1998). *Quantifiable data mining using principal component analysis* (Tech. Rep.).

Laurikkala, J., Juhola, M., Kentala, E., Lavrac, N., Miksch, S., & Kavsek, B. (2000). Informal identification of outliers in medical data. In *Fifth international workshop on intelligent data analysis in medicine and pharmacology* (Vol. 1, pp. 20–24).

Liu, J., Liu, P., Liu, S., Ma, Y., & Yang, W. (2013). Handover optimization in business processes via prediction. *Kybernetes*, *42*, 1101-1127.

Mans, R., Schonenberg, H., Leonardi, G., Panzarasa, S., Cavallini, A., Quaglini, S., & van der AALST, W. (2008). Process mining techniques: an application to stroke care. *Studies in health technology and informatics*, *136*, 573.

Nakatumba, J., & van der Aalst, W. M. P. (2009). Analyzing resource behavior using process mining. In *Business process management workshops.*

Online Courses 1.1 overview of time series characteristics. (n.d.). https://onlinecourses.science.psu.edu/stat510/node/47/. (Accessed: 09-08-2018)

Pearson, K. (1895). Contributions to the mathematical theory of evolution. ii. skew variation in homogeneous material. *Philosophical transactions of the Royal Society of London*, *186*(Part I), 343–424.

Pqsystems histogram: Study the shape. (n.d.). http://www.pqsystems.com/qualityadvisor/DataAnalysisTools/inter (Accessed: 07-05-2018)

Ramaswamy, S., Rastogi, R., & Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. In *Acm sigmod record* (Vol. 29, pp. 427–438).

Robbins, N. (2012). *Forbes a histogram is not a bar chart.* https://www.forbes.com/sites/naomirobbins/2012/01/04/a-histogram-is-not-a-bar-chart/5cd69e566d77 (Accessed: 07-05-2018)

Roberts, S., & Tarassenko, L. (1994). A probabilistic resource allocating network for novelty detection. *Neural Computation*, *6*(2), 270–284.

Roberts, S. J. (1999). Novelty detection using extreme value statistics. *IEE Proceedings-Vision, Image and Signal Processing*, *146*(3), 124–129.

Rodrigues, A. M. B., Almeida, C. F. P., Saraiva, D. D. G., Moreira, F. B., Spyrides, G. M., Varela, G., ... Fernandes, W. P. D. (2017). Starway to value: Mining a loan application process. *BPIC 2017.*

Rozinat, A. (n.d.). *Fluxicon: Disco user's guide.*

Santoyo, S. (2017). *Towards Data Science a brief overview of outlier detection techniques.* https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561. (Accessed: 23-04-2018)

Senderovich, A. (2017). Queue mining: Service perspectives in process mining. In *Bpm.*

Senderovich, A., Weidlich, M., & Gal, A. (2017). Feature learning for accurate time prediction in congested healthcare systems..

Senderovich, A., Weidlich, M., Gal, A., & Mandelbaum, A. (2014). Queue mining - predicting delays in service processes. In *Caise.*

Shearer, C. (2000). The crisp-dm model: the new blueprint for data mining. *Journal of data ware-*

housing, *5*(4), 13–22.

Sheynin, O. B. (1966). Origin of the theory of errors. *Nature*, *211*(5052), 1003.

Stack overflow peak signal detection in realtime timeseries data. (n.d.). `https://stackoverflow.com/questions/22583391/peak-signal-detection-in-realtime-timeseries-data`. (Accessed: 23-04-2018)

Sturges, H. A. (1926). The choice of a class interval. *Journal of the american statistical association*, *21*(153), 65–66.

Suchy, J., & Suchy, M. (2014). Predictive model for supporting itil business change management processes. *BPIC 2014*.

Susanti, Y., Pratiwi, H., et al. (2014). M estimation, s estimation, and mm estimation in robust regression. *International Journal of Pure and Applied Mathematics*, *91*(3), 349–360.

Teinemaa, I., Leontjeva, A., & Masing, K.-O. (2015). Bpic 2015: Diagnostics of building permit application process in dutch municipalities. *BPIC 2015*.

ter Hofstede, A. H., van der Aalst, W. M., Adams, M., & Russell, N. (2009). *Modern business process automation: Yawl and its support environment.* Springer Science & Business Media.

Van Der Aalst, W. (2012). Process mining: Overview and opportunities. *ACM Transactions on Management Information Systems (TMIS)*, *3*(2), 7.

Van der Aalst, W. M. (2016). *Process mining: data science in action.* Springer.

van der Aalst, W. M., & Stahl, C. (2011). *Modeling business processes: a petri net-oriented approach.* MIT press.

Verbeek, M. (2008). *A guide to modern econometrics.* John Wiley & Sons.

Verenich, I., Nguyen, H., Rosa, M. L., & Dumas, M. (2017). White-box prediction of process performance indicators via flow analysis. In *Icssp.*

Weller, T., & Maleshkova, M. (2017). Activity duration prediction of workflows by using a data science approach: Unveiling the advantage of semantics. In *Swit@iswc.*

Yu, C., & Yao, W. (2017). Robust linear regression: A review and comparison. *Communications in Statistics-Simulation and Computation*, *46*(8), 6261–6282.

Appendix

A More information on the datasets

The number of days that are in an event log are shown in table 7 and the number of events in every event log is shown in table 8.

Table 7: Number of days in a dataset

Event Log	Start Date	End Date	Number of days
Sepsis	07-11-2013	05-06-2015	575
HL	03-01-2005	20-03-2008	1172
BPIC 2012	01-10-2011	14-03-2012	165
BPIC 2107	01-01-2016	31-12-2017	730
RTF	01-01-2000	18-06-2013	4917
HB	13-12-2012	19-01-2016	1132

Table 8: Number of cases and events

Event Log	Cases	Events
Sepsis	1.050	15.214
HL	1.143	150.291
BPIC 2012	13.087	262.200
BPIC 2107	31.509	561.671
RTF	150.370	561.470
HB	179.060	451.359

B Validation time series classification method

Table 9 until 16 give the visual and automatic categories of all time series that were tested.

Table 9: Events Per Day

Data set	**Visual**	**Automatic**
Sepsis	Random	Random
HL	Weekly Pattern and Clear Peaks	Weekly Pattern and Clear Peaks
BPIC 2012	Weekly Pattern	Weekly Pattern and Clear Peaks
BPIC 2017	Weekly Pattern	Weekly Pattern and Clear Peaks
RTF	Clear Peaks	Weekly Pattern and Clear Peaks
HB	Smooth	Smooth

Table 10: Number of Case Variants on a Day

Data set	**Visual**	**Automatic**
Sepsis	Random	Random
HL	Weekly Pattern	Weekly Pattern
BPIC 2012	Weekly Pattern	Weekly Pattern and Clear Peaks
BPIC 2017	Weekly Pattern and Clear Peaks	Weekly Pattern and Clear Peaks
RTF	Random	Random
HB	Smooth	Random

For the duration of activities we did not only look at the complete dataset over time, but also at the duration of the separate activities over time. This is shown in the following table (table 11). The visual conclusion for the duration of activities over time when looking at all activities together, this is that the category for this time series is Clear peaks, while the automatic categorisation says it is Weekly pattern and Clear peaks.

Table 11: Duration of activities

Activity	**Visual**	**Automatic**
A_Create Application	None*	Few Values
A_Submitted	None*	Few Values
A_Concept	None*	Few Values
W_Complete Application	Clear Peaks	Weekly Pattern and Clear Peaks
A_Accepted Offer	None*	Few Values
O_Create Offer	None*	Few Values
O_Created	None*	Few Values
O_Sent (Mail and Online)	None*	Few Values
W_Call After Offers	Clear Peaks	Clear Peaks
A_Complete	None*	Few Values
W_Validate Application	Random	Smooth
A_Validating	None*	Few Values
O_Returned	None*	Few Values
W_Call Incomplete Files	Clear Peaks	Weekly Pattern
A_Incomplete	None*	Few Values
O_Accepted	None*	Few Values
A_Pending	None*	Few Values
A_Denied	None*	Few Values
O_Refused	None*	Few Values
O_Cancelled	None*	Few Values
W_Handle Leads	Clear Peaks	Smooth
A_Cancelled	None*	Few Values
O_Sent (Online Only)	None*	Few Values
W_Asses Potential Fraud	Clear Peaks	Weekly Pattern
W_Personal Loan Collection	None*	Few Values
W_Shortened Completion	None*	Few Values

*None implies that every value of the time series is zero.

Table 12: Duration of cases

Data set	**Visual**	**Automatic**
Sepsis	Random	Random
HL	Random	Weekly Pattern
BPIC 2012	Smooth	Weekly Pattern
BPIC 2017	Clear Peaks	Smooth
RTF	Random	Random
HB	Smooth	Smooth

The last set of time series that was made looks at the average duration on a day if a certain activity is performed on a day. The results of the visual and automatic categorisation of these time series is shown in table 13 until 16. These time series are, however, only used for validation of the methods.

Table 13: Duration of cases per activity Sepsis

Activity	**Visual**	**Automatic**
ER Registration	Random	Random
Leucocytes	Clear Peaks	Random
CRP	Random	Random
LacticAcid	Clear Peaks	Clear Peaks
ER Triage	Random	Random
ER Sepsis Triage	Random	Random
IV Liquid	Random	Random
IV Antibiotics	Random	Random
Admission NC	Clear Peaks	Random
Release A	Random	Random
Return ER	Random	Random
Admission IC	Random	Weekly Pattern
Release B	Few Values	Weekly Pattern and Clear Peaks
Release C	Few Values	Few Values
Release D	Few Values	Few Values
Release E	Few Values	Few Values

Table 14: Duration of cases per activity BPIC 2012

Activity	**Visual**	**Automatic**
A_Submitted-Complete	Smooth	Smooth
A_PartlySubmitted-Complete	Smooth	Smooth
A_Preaccepted-Complete	Smooth	Clear Peaks
W_Completeren Aanvraag-Schedule	Smooth	Random
W_Completeren Aanvraag-Start	Weekly Pattern	Weekly Pattern and Clear Peaks
A_Accepted-Complete	Weekly Pattern	Weekly Pattern and Clear Peaks
O_Selected-Complete	Weekly Pattern	Weekly Pattern and Clear Peaks
A_Finalized-Complete	Weekly Pattern	Weekly Pattern and Clear Peaks
O_Created-Complete	Weekly Pattern	Weekly Pattern and Clear Peaks
O_Sent-Complete	Weekly Pattern	Weekly Pattern and Clear Peaks
W_Nabellen Offertes-Schedule	Weekly Pattern	Weekly Pattern and Clear Peaks
W_Completeren Aanvraag-Complete	Clear Peaks	Clear Peaks
W_Nabellen Offertes-Start	Weekly Pattern	Weekly Pattern and Clear Peaks
W_Nabellen Offertes-Complete	Clear Peaks	Clear Peaks
O_Sent Back-Complete	Weekly Pattern	Weekly Pattern
W_Valideren Aanvraag-Schedule	Weekly Pattern and Clear Peaks	Weekly Pattern and Clear Peaks
W_Valideren Aanvraag-Start	Weekly Pattern and Clear Peaks	Weekly Pattern and Clear Peaks
A_Registered-Complete	Weekly Pattern	Weekly Pattern
A_Approved-Complete	Weekly Pattern	Weekly Pattern
O_Accepted-Complete	Weekly Pattern	Weekly Pattern
A_Activated-Complete	Weekly Pattern	Weekly Pattern
W_Valideren Aanvraag-Complete	Weekly Pattern and Clear Peaks	Weekly Pattern and Clear Peaks
O_Cancelled-Complete	Clear Peaks	Clear Peaks
W_Wijzigen Contractgegevens-Schedule	Few Values	Few Values
A_Declined-Complete	Weekly Pattern and Clear Peaks	Clear Peaks
A_Cancelled-Complete	Clear Peaks	Weekly Pattern and Clear Peaks
W_Afhandelen Leads-Schedule	Clear Peaks	Clear Peaks
W_Afhandelen Leads-Start	Weekly Pattern and Clear Peaks	Weekly Pattern and Clear Peaks
W_Afhandelen Leads-Complete	Weekly Pattern and Clear Peaks	Weekly Pattern and Clear Peaks
O_Declined-Complete	Weekly Pattern and Clear Peaks	Weekly Pattern and Clear Peaks
W_Nabellen Incomplete Dossiers-Schedule	Weekly Pattern	Weekly Pattern
W_Nabellen Incomplete Dossiers-Start	Weekly Pattern and Clear Peaks	Weekly Pattern and Clear Peaks
W_Nabellen Incomplete Dossiers-Complete	Weekly Pattern and Clear Peaks	Weekly Pattern and Clear Peaks
W_Beoordelen Fraude-Schedule	Random	Weekly Pattern and Clear Peaks
W_Beoordelen Fraude-Start	Random	Weekly Pattern and Clear Peaks
W_Beoordelen Fraude-Complete	Random	Weekly Pattern and Clear Peaks

Table 15: Duration of cases per activity RTF

Activity	**Visual**	**Automatic**
Create Fine	Random	Random
Send Fine	Clear Peaks	Random
Insert Fine Notification	Random	Random
Add Penalty	Clear Peaks	Clear Peaks
Send for Credit Collection	Few Values	Few Values
Payment	Clear Peaks	Weekly Patterns
Insert Date Appeal to Prefecture	Random	Weekly Patten and Clear Peaks
Send Appeal to Prefecture	Clear Peaks	Weekly Patten and Clear Peaks
Receive Result Appeal from Prefecture	Random	Weekly Pattern
Notify Result Appeal to Offender	Random	Weekly Pattern
Appeal to Judge	Random	Weekly Pattern

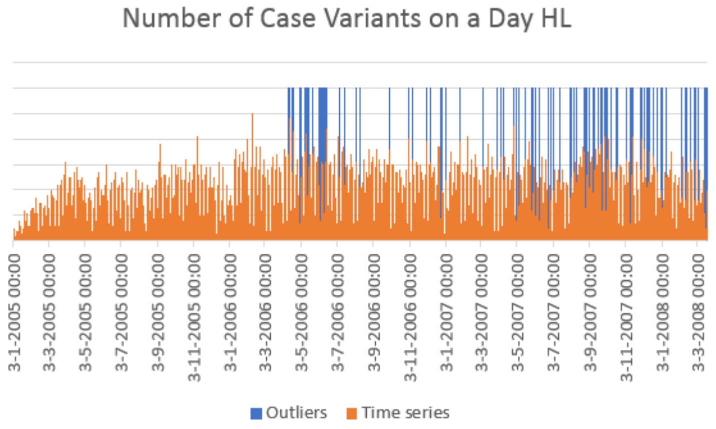
Table 16: Duration of cases per activity HB

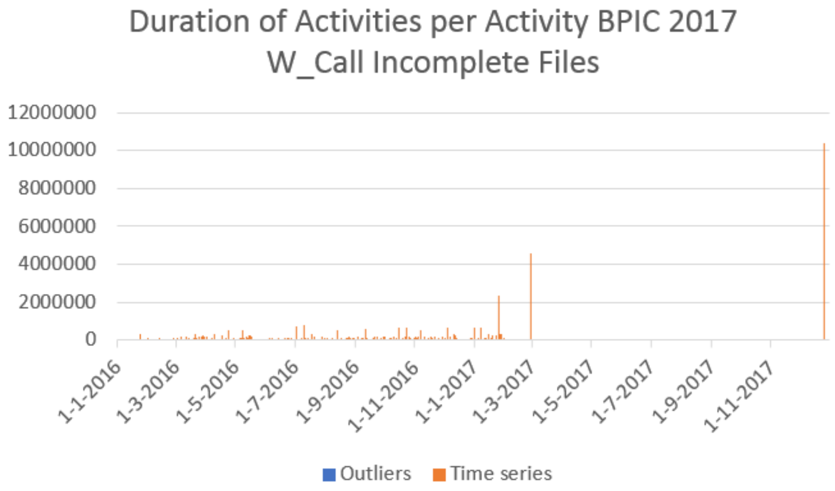
Activity	**Visual**	**Automatic**
New	Smooth	Random
Fin	Smooth	Random
Release	Smooth	Random
Code OK	Random	Random
Billed	Random	Random
Delete	Random	Random
Reopen	Random	Random
Change Diagn	Smooth	Smooth
Storno	Random	Clear Peaks
Reject	Random	Clear Peaks
Code NOK	Clear Peaks	Clear Peaks
Set Status	Random	Clear Peaks
Change End	Few Values	Few Values
Manual	Random	Clear Peaks
Join-Pat	Few Values	Weekly Pattern
Code Error	Few Values	Random
ZDBC-Behan	Few Values	Few Values
Empty	Random	Clear Peaks

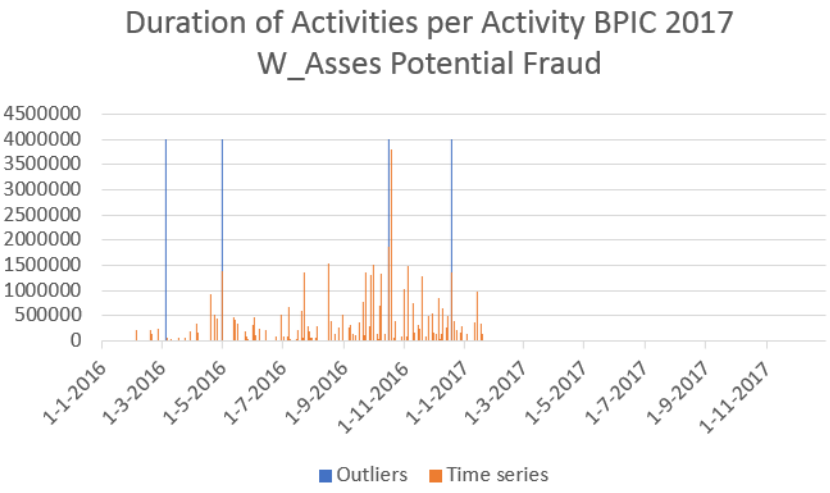
C Validation outlier detection algorithms

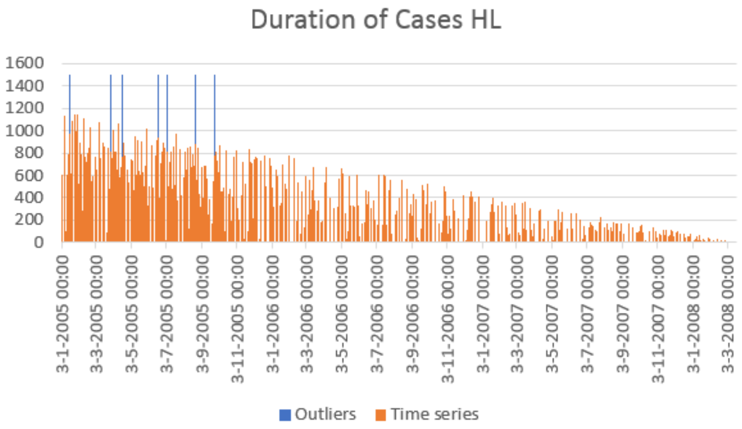
C.1 Weekly pattern

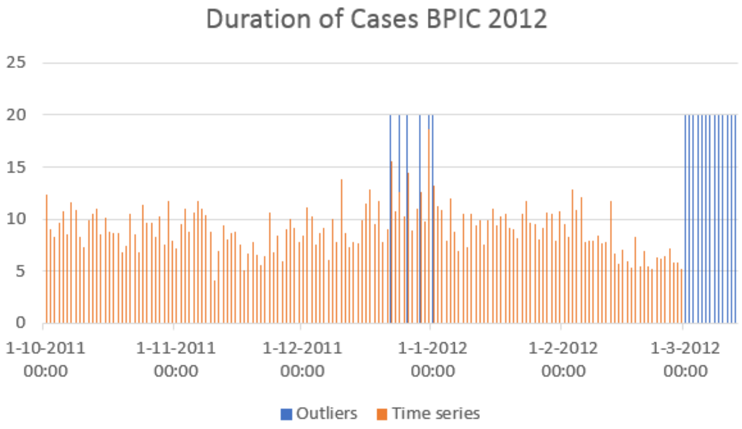
C.1.1 Only Weekly pattern

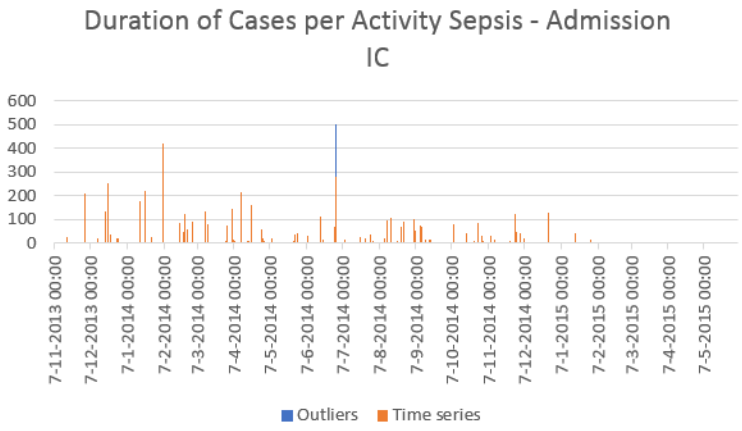


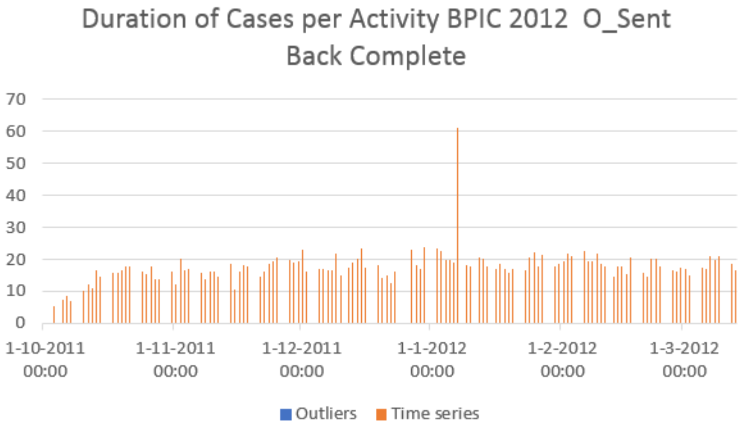


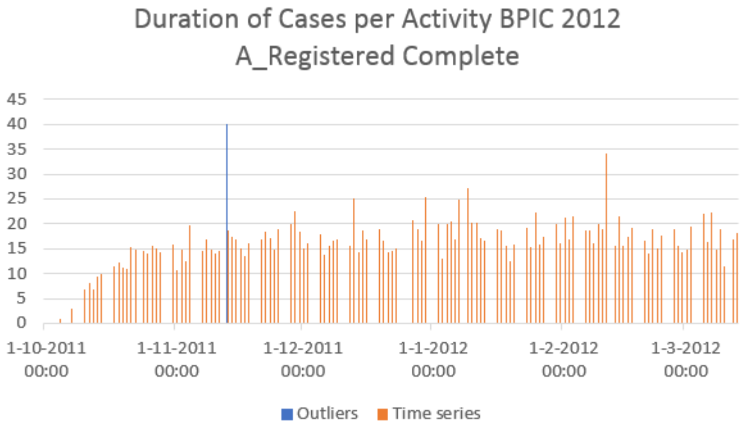

Duration of Activities per Activity BPIC 2017
W_Asses Potential Fraud
4500000
4000000
3500000
3000000
2500000
2000000
1500000
1000000
500000
0
1-1-2016
1-3-2016
1-5-2016
1-7-2016
1-9-2016
1-11-2016
1-1-2017
1-3-2017
1-5-2017
1-7-2017
1-9-2017
1-11-2017
Outliers
Time series

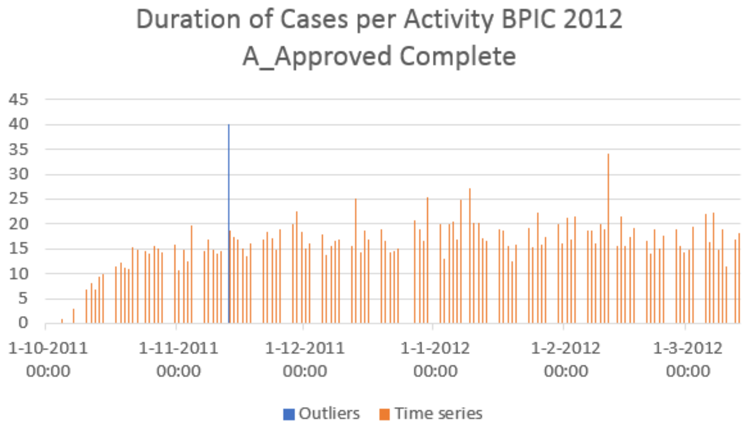

Duration of Cases HL
1600
1400
1200
1000
800
600
400
200
0
3-1-2005 00:00
3-3-2005 00:00
3-5-2005 00:00
3-7-2005 00:00
3-9-2005 00:00
3-11-2005 00:00
3-1-2006 00:00
3-3-2006 00:00
3-5-2006 00:00
3-7-2006 00:00
3-9-2006 00:00
3-11-2006 00:00
3-1-2007 00:00
3-3-2007 00:00
3-5-2007 00:00
3-7-2007 00:00
3-9-2007 00:00
3-11-2007 00:00
3-1-2008 00:00
3-3-2008 00:00
Outliers
Time series

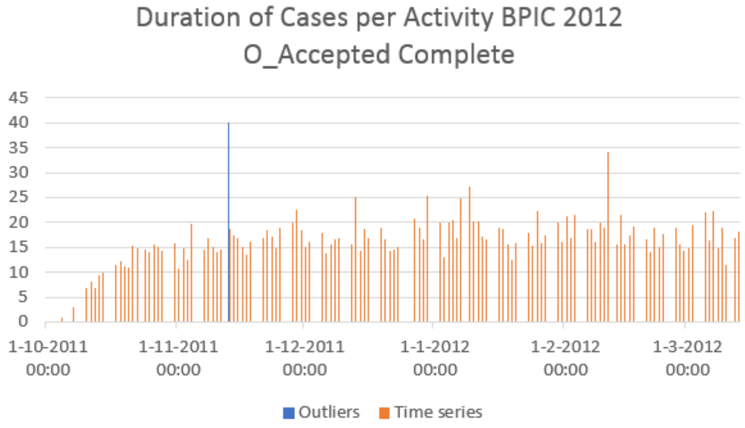

Duration of Cases BPIC 2012
25
20
15
10
5
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series

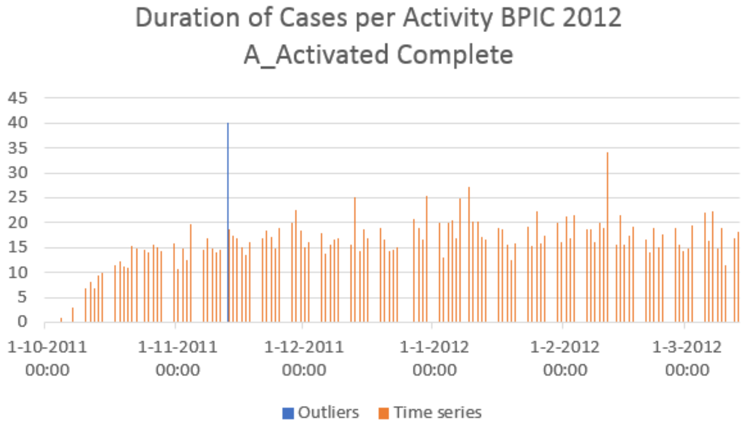

Duration of Cases per Activity Sepsis - Admission IC
600
500
400
300
200
100
0
7-11-2013 00:00
7-12-2013 00:00
7-1-2014 00:00
7-2-2014 00:00
7-3-2014 00:00
7-4-2014 00:00
7-5-2014 00:00
7-6-2014 00:00
7-7-2014 00:00
7-8-2014 00:00
7-9-2014 00:00
7-10-2014 00:00
7-11-2014 00:00
7-12-2014 00:00
7-1-2015 00:00
7-2-2015 00:00
7-3-2015 00:00
7-4-2015 00:00
7-5-2015 00:00
Outliers
Time series

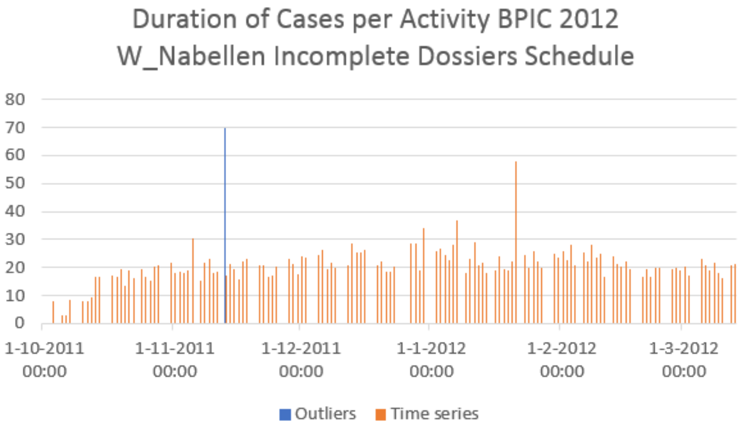

Duration of Cases per Activity BPIC 2012 O_Sent Back Complete
70
60
50
40
30
20
10
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series

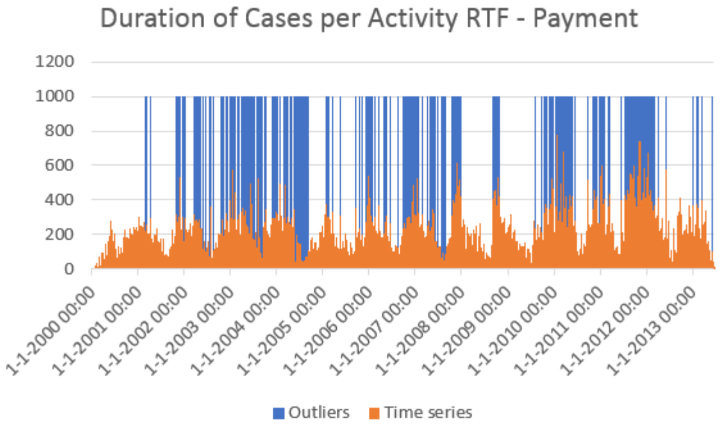

Duration of Cases per Activity BPIC 2012 A_Registered Complete
45
40
35
30
25
20
15
10
5
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series

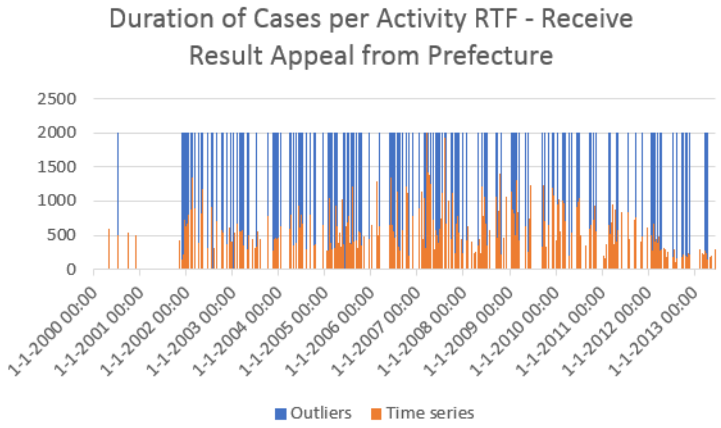

Duration of Cases per Activity BPIC 2012
A_Approved Complete
45
40
35
30
25
20
15
10
5
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
■ Outliers
■ Time series

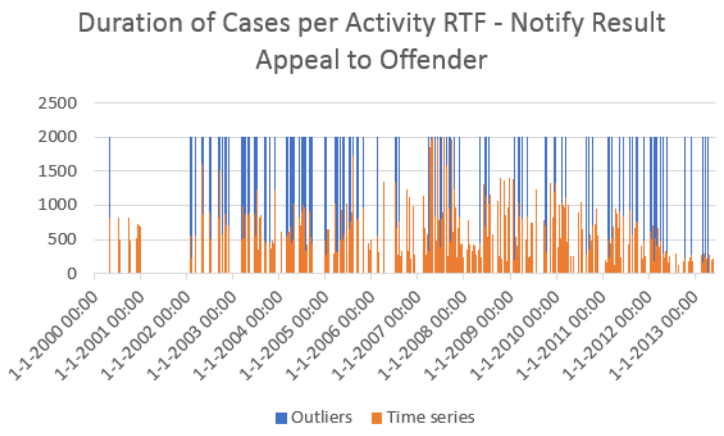

Duration of Cases per Activity BPIC 2012
O_Accepted Complete
45
40
35
30
25
20
15
10
5
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
■ Outliers
■ Time series

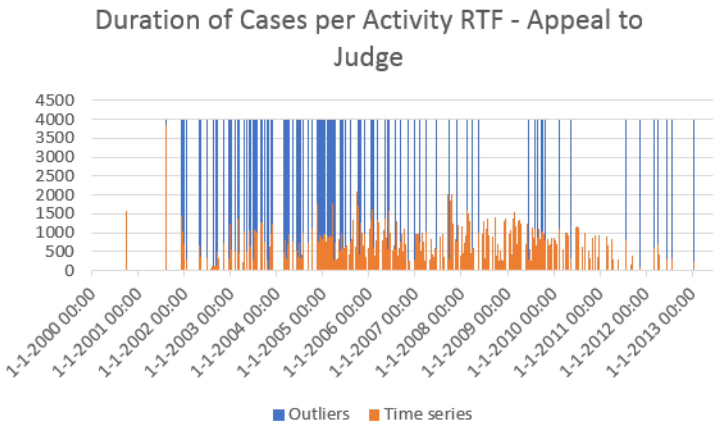

Duration of Cases per Activity BPIC 2012
A_Activated Complete
45
40
35
30
25
20
15
10
5
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
■ Outliers
■ Time series

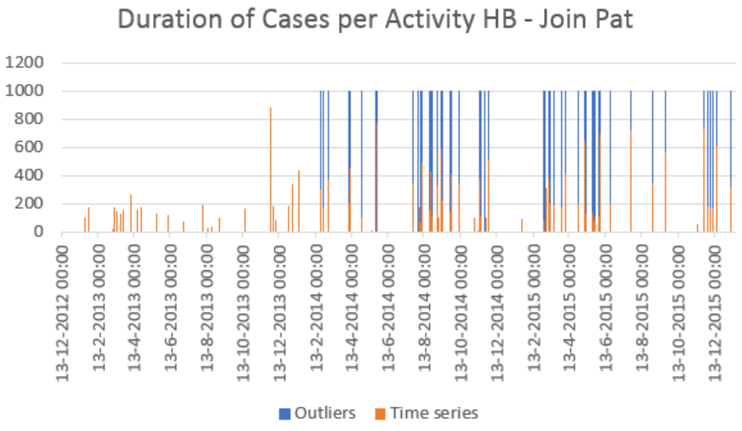

Duration of Cases per Activity BPIC 2012
W_Nabellen Incomplete Dossiers Schedule
80
70
60
50
40
30
20
10
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series


Duration of Cases per Activity RTF - Payment
1200
1000
800
600
400
200
0
1-1-2000 00:00
1-1-2001 00:00
1-1-2002 00:00
1-1-2003 00:00
1-1-2004 00:00
1-1-2005 00:00
1-1-2006 00:00
1-1-2007 00:00
1-1-2008 00:00
1-1-2009 00:00
1-1-2010 00:00
1-1-2011 00:00
1-1-2012 00:00
1-1-2013 00:00
Outliers
Time series

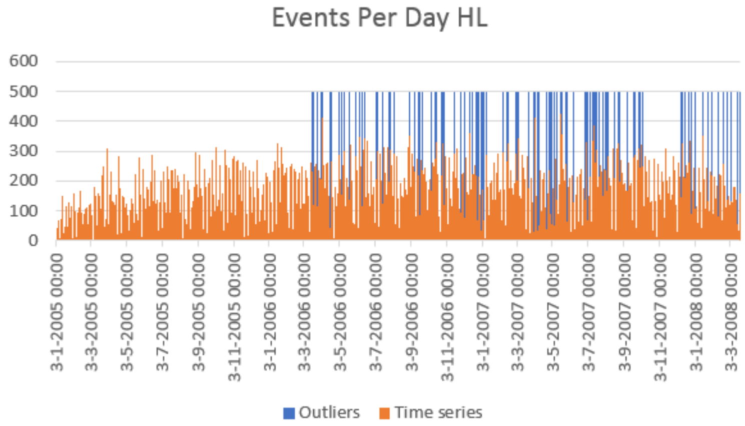

Duration of Cases per Activity RTF - Receive
Result Appeal from Prefecture
2500
2000
1500
1000
500
0
1-1-2000 00:00
1-1-2001 00:00
1-1-2002 00:00
1-1-2003 00:00
1-1-2004 00:00
1-1-2005 00:00
1-1-2006 00:00
1-1-2007 00:00
1-1-2008 00:00
1-1-2009 00:00
1-1-2010 00:00
1-1-2011 00:00
1-1-2012 00:00
1-1-2013 00:00
Outliers
Time series

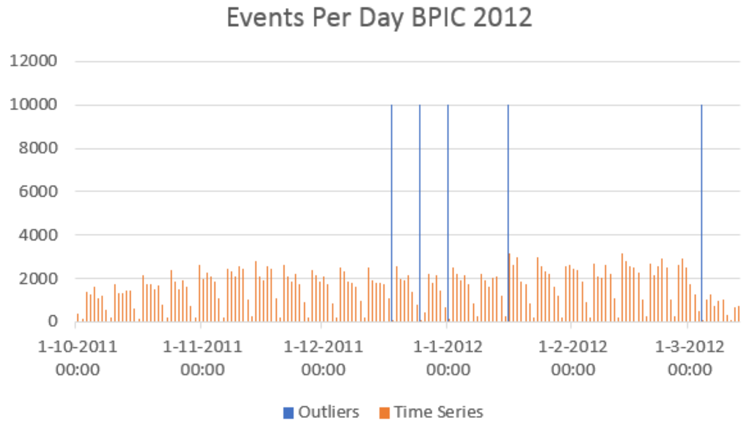

Duration of Cases per Activity RTF - Notify Result Appeal to Offender
2500
2000
1500
1000
500
0
1-1-2000 00:00
1-1-2001 00:00
1-1-2002 00:00
1-1-2003 00:00
1-1-2004 00:00
1-1-2005 00:00
1-1-2006 00:00
1-1-2007 00:00
1-1-2008 00:00
1-1-2009 00:00
1-1-2010 00:00
1-1-2011 00:00
1-1-2012 00:00
1-1-2013 00:00
Outliers
Time series

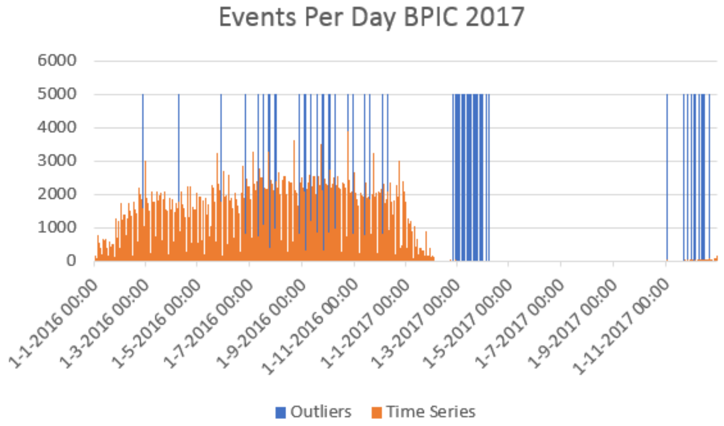

Duration of Cases per Activity RTF - Appeal to Judge
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1-1-2000 00:00
1-1-2001 00:00
1-1-2002 00:00
1-1-2003 00:00
1-1-2004 00:00
1-1-2005 00:00
1-1-2006 00:00
1-1-2007 00:00
1-1-2008 00:00
1-1-2009 00:00
1-1-2010 00:00
1-1-2011 00:00
1-1-2012 00:00
1-1-2013 00:00
Outliers
Time series

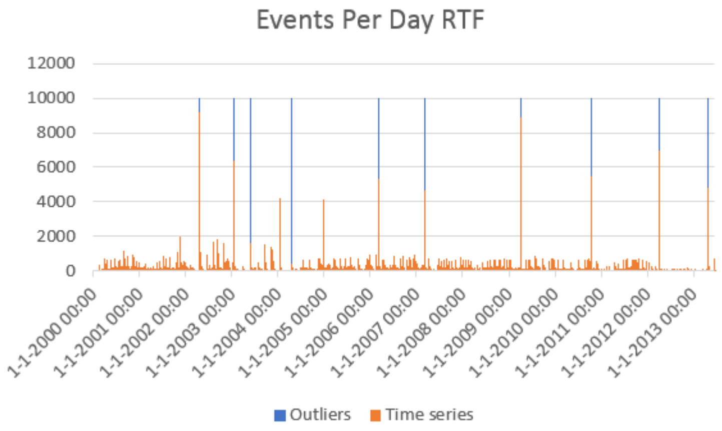

Duration of Cases per Activity HB - Join Pat
1200
1000
800
600
400
200
0
13-12-2012 00:00
13-2-2013 00:00
13-4-2013 00:00
13-6-2013 00:00
13-8-2013 00:00
13-10-2013 00:00
13-12-2013 00:00
13-2-2014 00:00
13-4-2014 00:00
13-6-2014 00:00
13-8-2014 00:00
13-10-2014 00:00
13-12-2014 00:00
13-2-2015 00:00
13-4-2015 00:00
13-6-2015 00:00
13-8-2015 00:00
13-10-2015 00:00
13-12-2015 00:00
Outliers
Time series

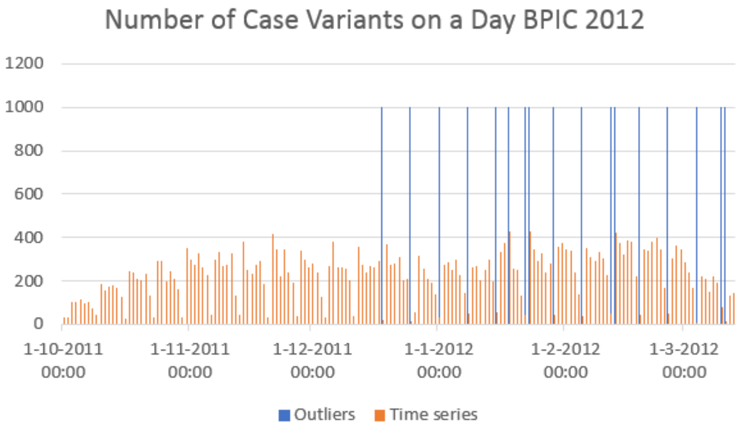
C.1.2 Weekly pattern and clear peaks

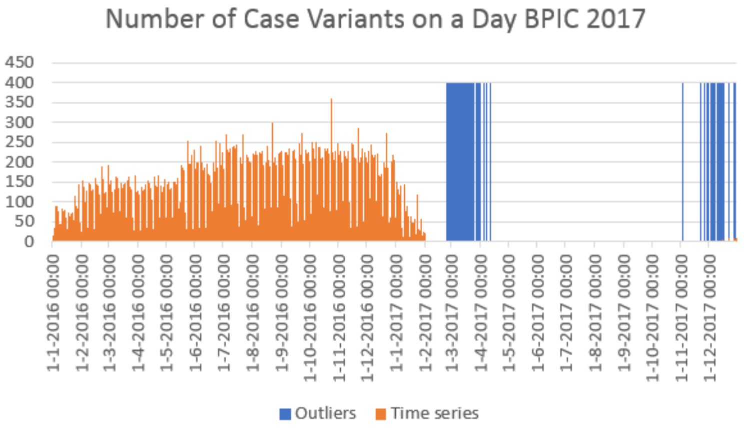


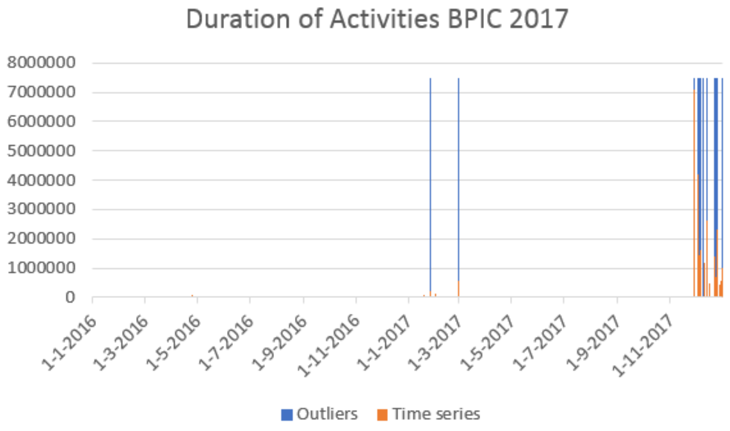


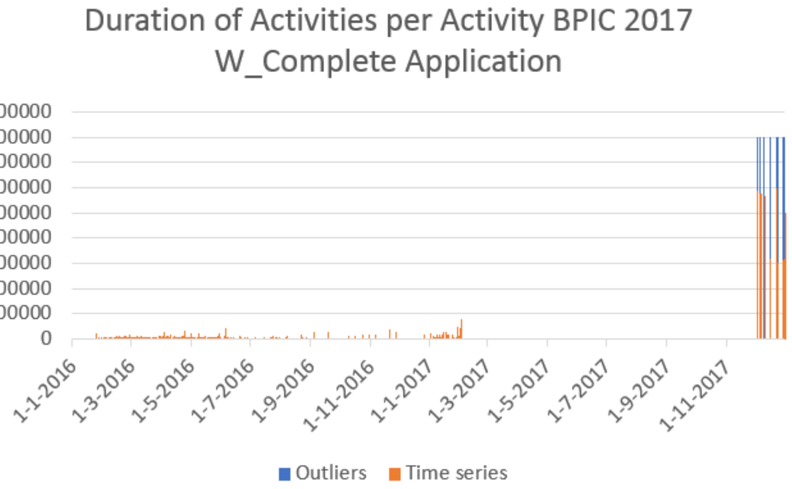


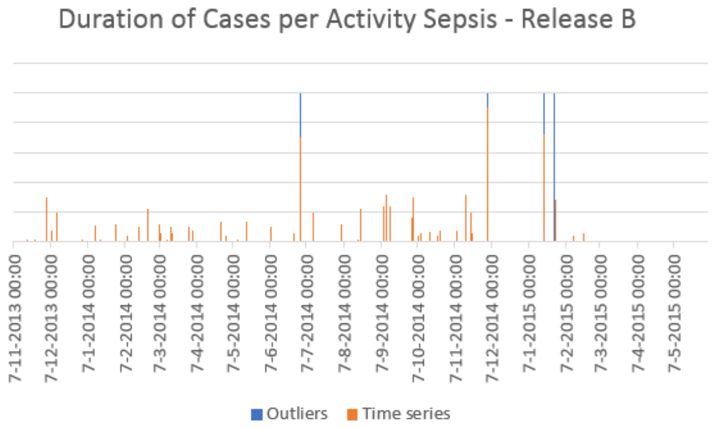

Events Per Day RTF
12000
10000
8000
6000
4000
2000
0
1-1-2000 00:00
1-1-2001 00:00
1-1-2002 00:00
1-1-2003 00:00
1-1-2004 00:00
1-1-2005 00:00
1-1-2006 00:00
1-1-2007 00:00
1-1-2008 00:00
1-1-2009 00:00
1-1-2010 00:00
1-1-2011 00:00
1-1-2012 00:00
1-1-2013 00:00
Outliers
Time series

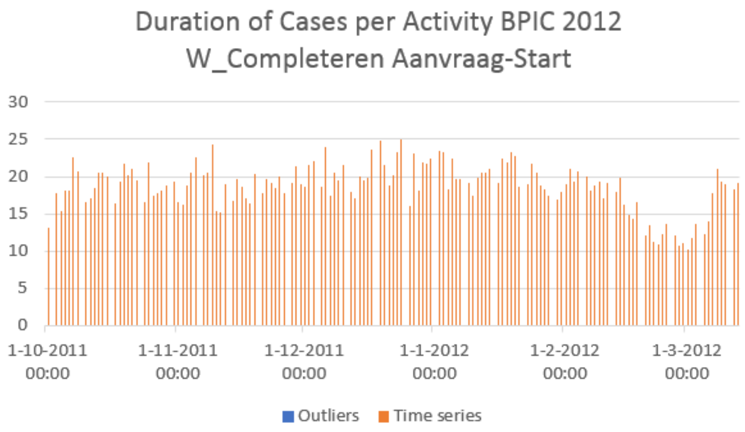

Number of Case Variants on a Day BPIC 2012
1200
1000
800
600
400
200
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series


Number of Case Variants on a Day BPIC 2017
450
400
350
300
250
200
150
100
50
0
1-1-2016 00:00
1-2-2016 00:00
1-3-2016 00:00
1-4-2016 00:00
1-5-2016 00:00
1-6-2016 00:00
1-7-2016 00:00
1-8-2016 00:00
1-9-2016 00:00
1-10-2016 00:00
1-11-2016 00:00
1-12-2016 00:00
1-1-2017 00:00
1-2-2017 00:00
1-3-2017 00:00
1-4-2017 00:00
1-5-2017 00:00
1-6-2017 00:00
1-7-2017 00:00
1-8-2017 00:00
1-9-2017 00:00
1-10-2017 00:00
1-11-2017 00:00
1-12-2017 00:00
Outliers
Time series

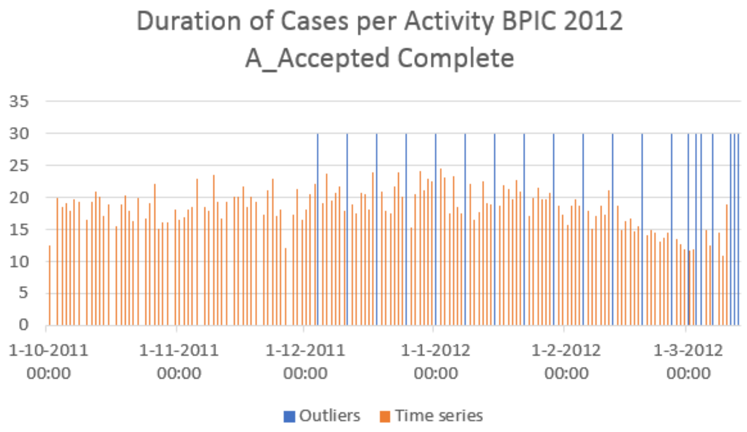

Duration of Activities BPIC 2017
8000000
7000000
6000000
5000000
4000000
3000000
2000000
1000000
0
1-1-2016
1-3-2016
1-5-2016
1-7-2016
1-9-2016
1-11-2016
1-1-2017
1-3-2017
1-5-2017
1-7-2017
1-9-2017
1-11-2017
Outliers
Time series


Duration of Activities per Activity BPIC 2017
W_Complete Application
45000000
40000000
35000000
30000000
25000000
20000000
15000000
10000000
5000000
0
1-1-2016
1-3-2016
1-5-2016
1-7-2016
1-9-2016
1-11-2016
1-1-2017
1-3-2017
1-5-2017
1-7-2017
1-9-2017
1-11-2017
Outliers
Time series

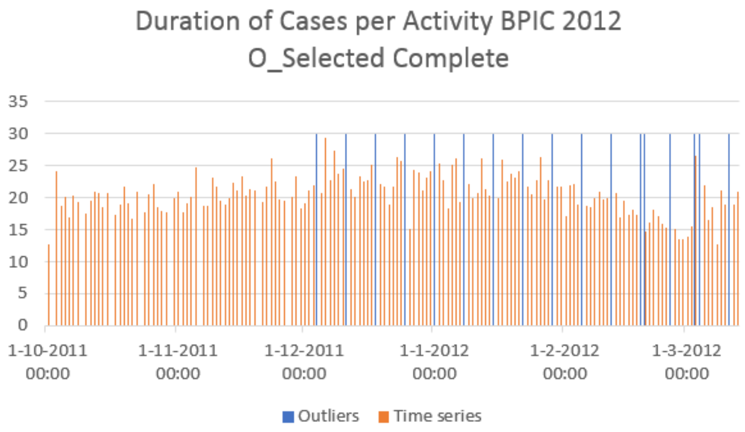

Duration of Cases per Activity Sepsis - Release B
60
50
40
30
20
10
0
7-11-2013 00:00
7-12-2013 00:00
7-1-2014 00:00
7-2-2014 00:00
7-3-2014 00:00
7-4-2014 00:00
7-5-2014 00:00
7-6-2014 00:00
7-7-2014 00:00
7-8-2014 00:00
7-9-2014 00:00
7-10-2014 00:00
7-11-2014 00:00
7-12-2014 00:00
7-1-2015 00:00
7-2-2015 00:00
7-3-2015 00:00
7-4-2015 00:00
7-5-2015 00:00
Outliers
Time series



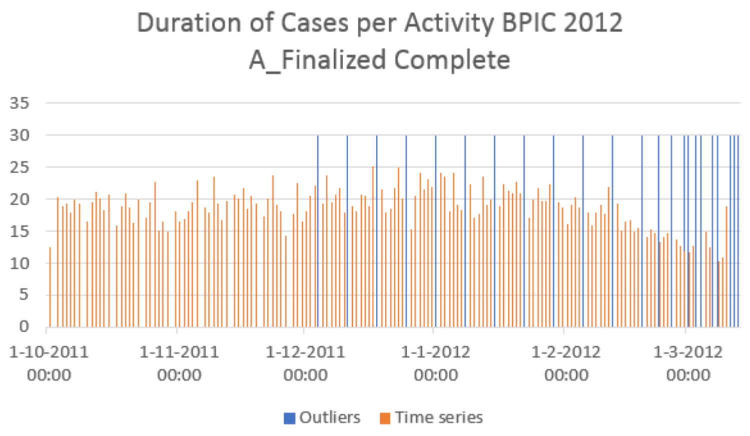
Duration of Cases per Activity BPIC 2012
W_Completeren Aanvraag-Start



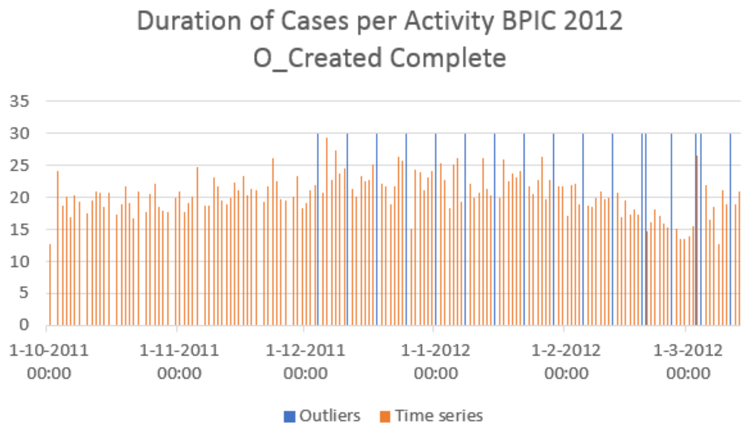
Duration of Cases per Activity BPIC 2012
A_Accepted Complete



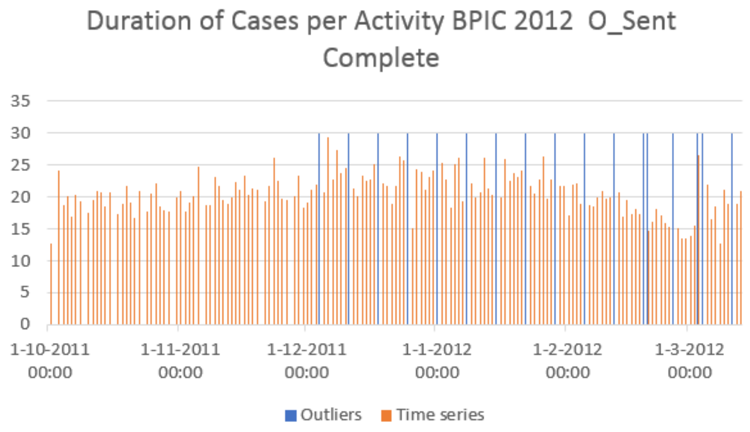
Duration of Cases per Activity BPIC 2012
O_Selected Complete



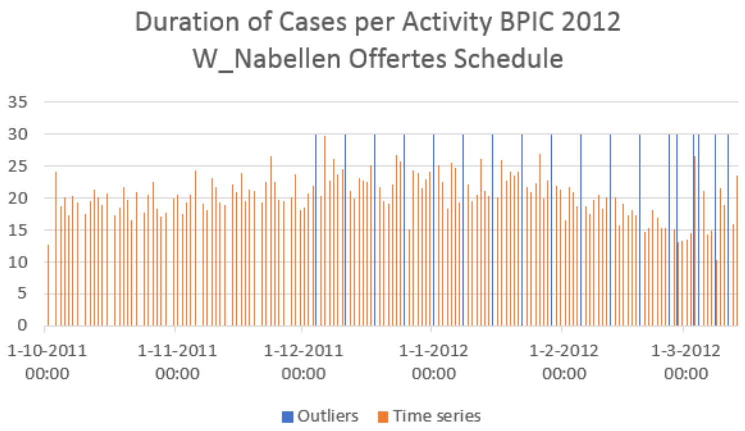
Duration of Cases per Activity BPIC 2012 A_Finalized Complete

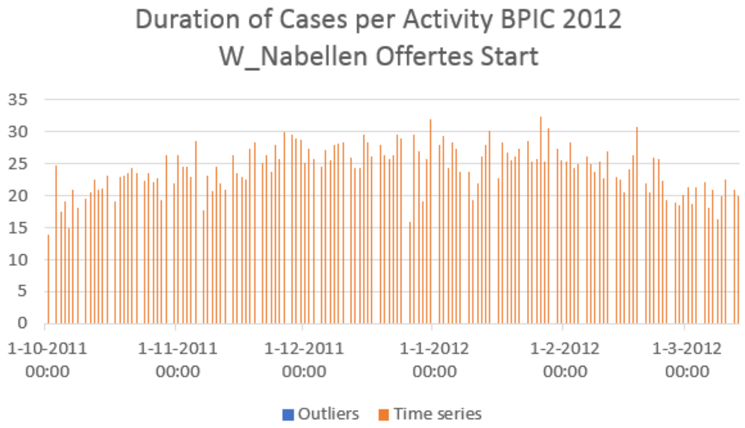


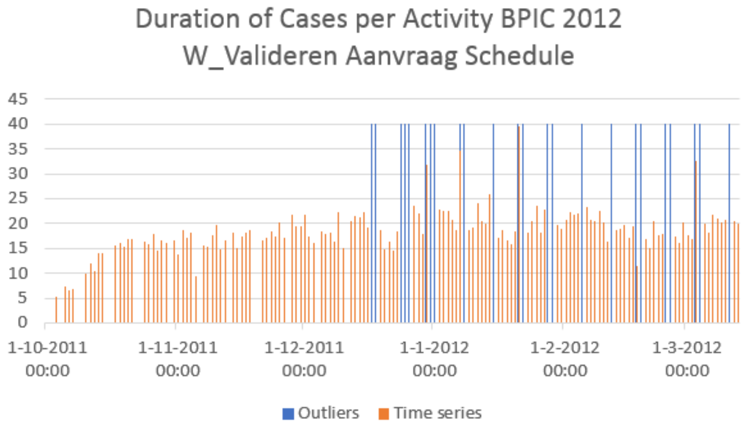
Duration of Cases per Activity BPIC 2012 O_Created Complete

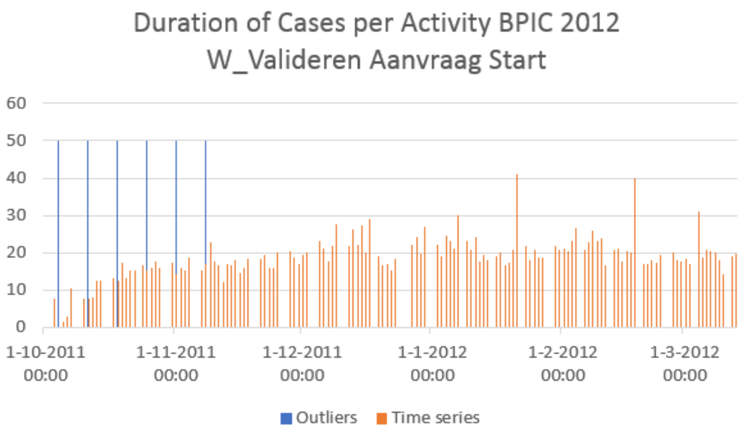


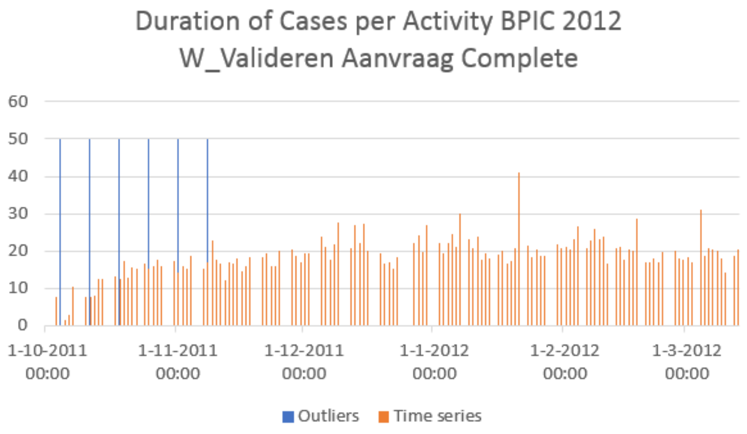
Duration of Cases per Activity BPIC 2012 O_Sent Complete

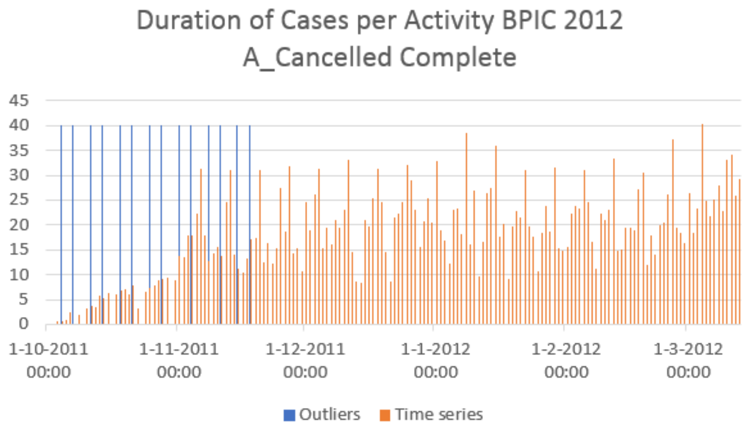

Duration of Cases per Activity BPIC 2012
W_Nabellen Offertes Schedule

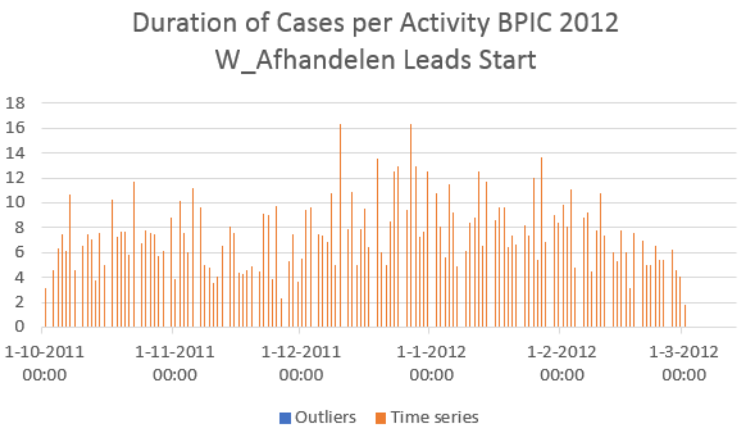

Duration of Cases per Activity BPIC 2012
W_Nabellen Offertes Start

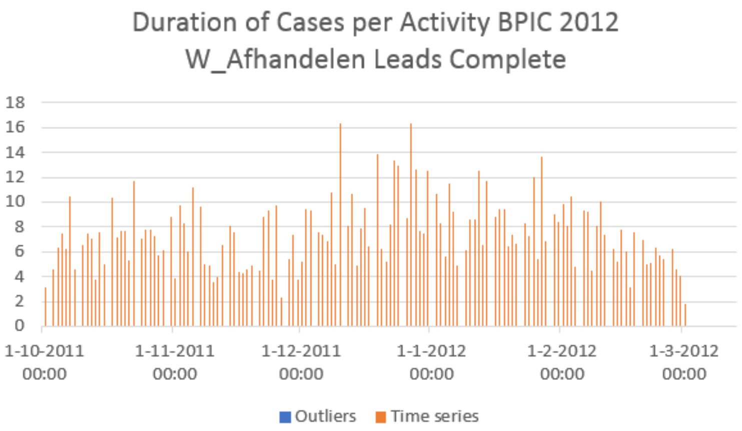

Duration of Cases per Activity BPIC 2012
W_Valideren Aanvraag Schedule

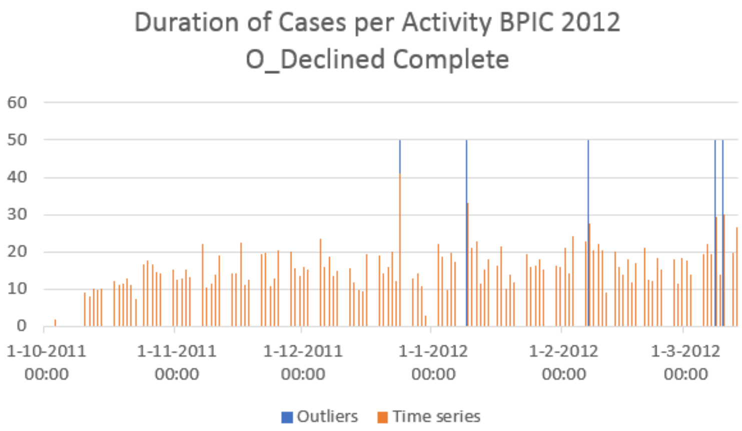

Duration of Cases per Activity BPIC 2012
W_Valideren Aanvraag Start
60
50
40
30
20
10
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series

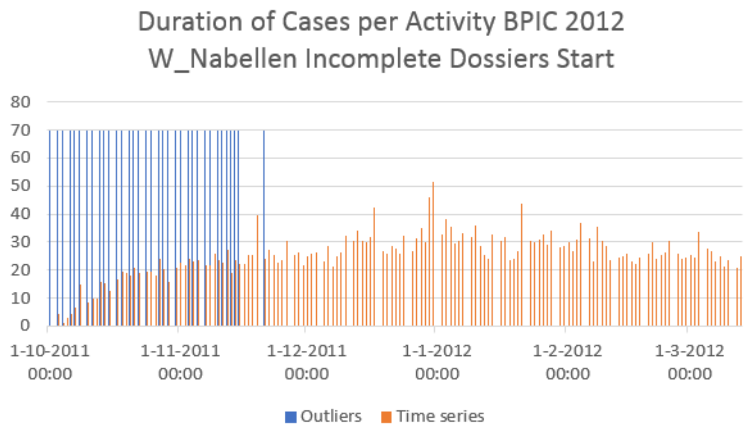

Duration of Cases per Activity BPIC 2012
W_Valideren Aanvraag Complete
60
50
40
30
20
10
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series

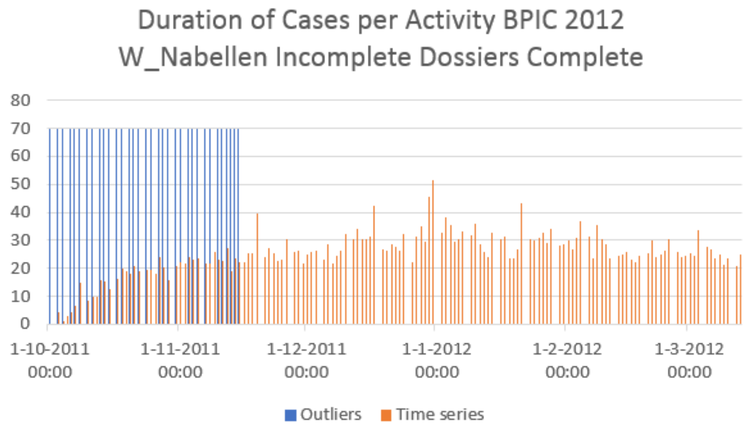

Duration of Cases per Activity BPIC 2012
A_Cancelled Complete
45
40
35
30
25
20
15
10
5
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series

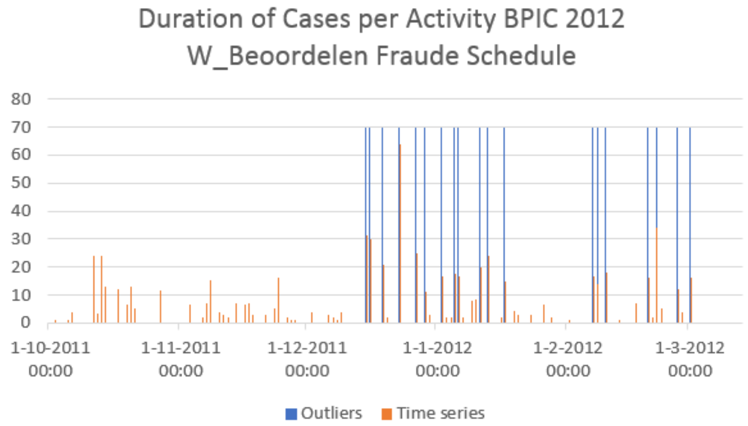

Duration of Cases per Activity BPIC 2012
W_Afhandelen Leads Start
18
16
14
12
10
8
6
4
2
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series

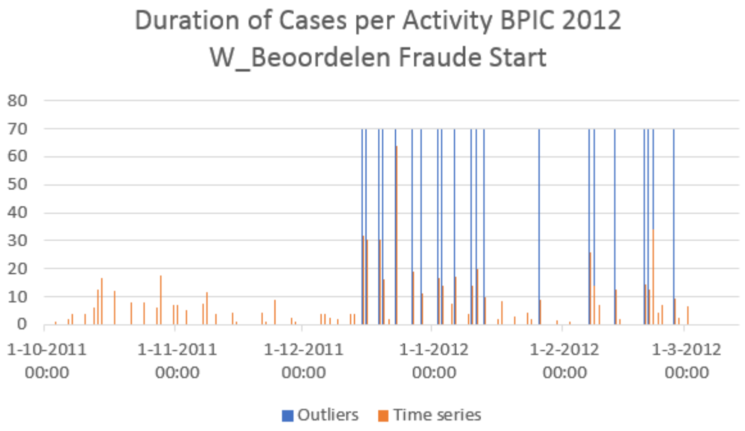

Duration of Cases per Activity BPIC 2012
W_Afhandelen Leads Complete
18
16
14
12
10
8
6
4
2
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series

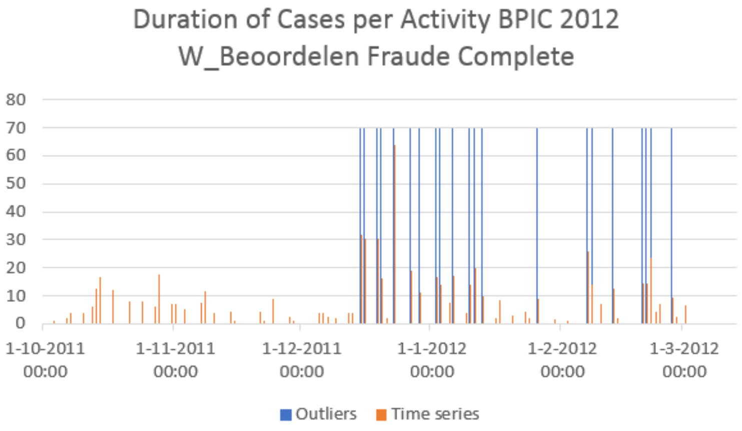

Duration of Cases per Activity BPIC 2012
O_Declined Complete
60
50
40
30
20
10
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series

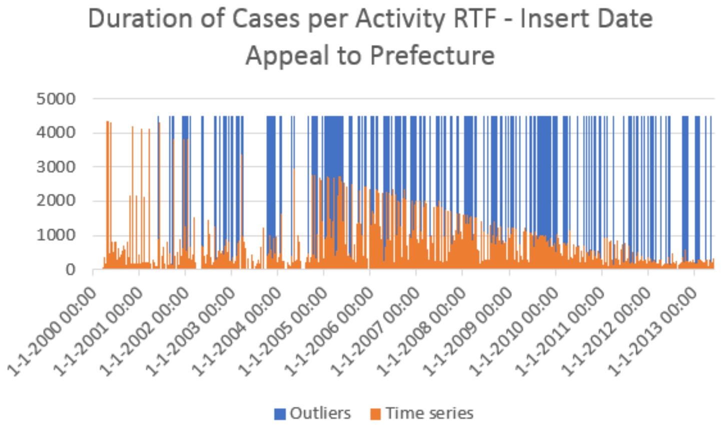

Duration of Cases per Activity BPIC 2012
W_Nabellen Incomplete Dossiers Start

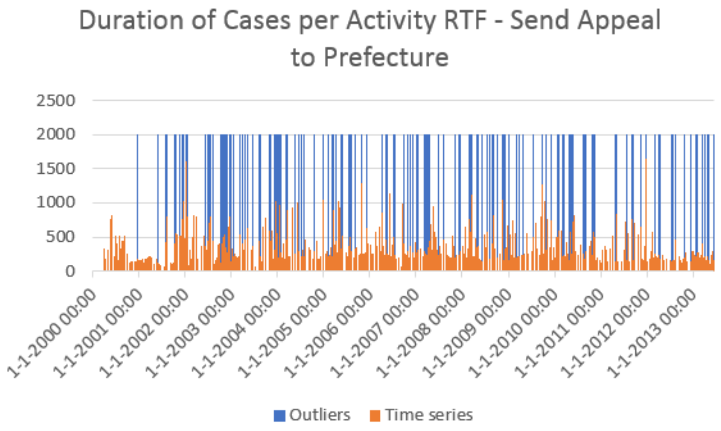

Duration of Cases per Activity BPIC 2012
W_Nabellen Incomplete Dossiers Complete


Duration of Cases per Activity BPIC 2012
W_Beoordelen Fraude Schedule

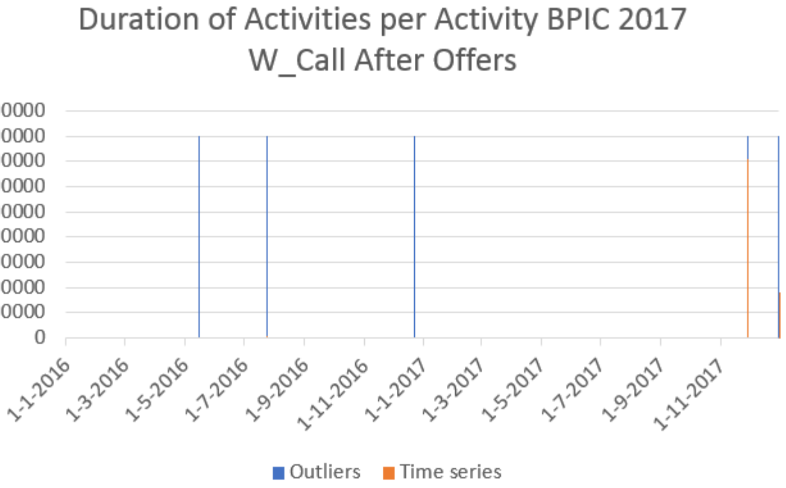

Duration of Cases per Activity BPIC 2012 W_Beoordelen Fraude Start

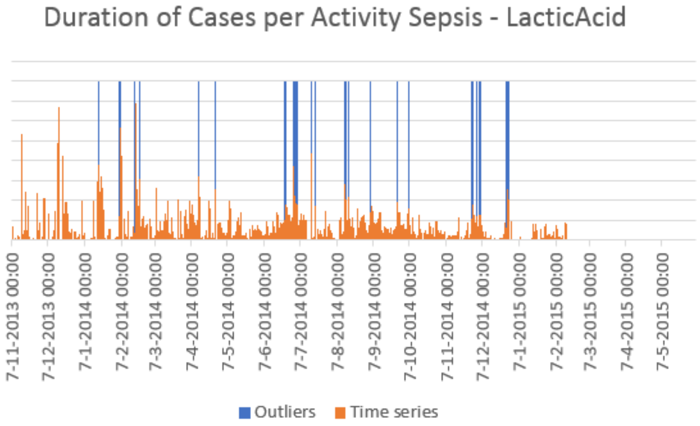

Duration of Cases per Activity BPIC 2012 W_Beoordelen Fraude Complete

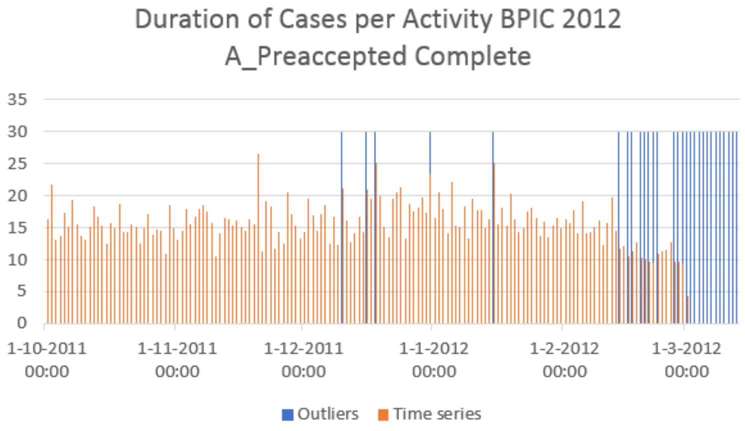

Duration of Cases per Activity RTF - Insert Date Appeal to Prefecture

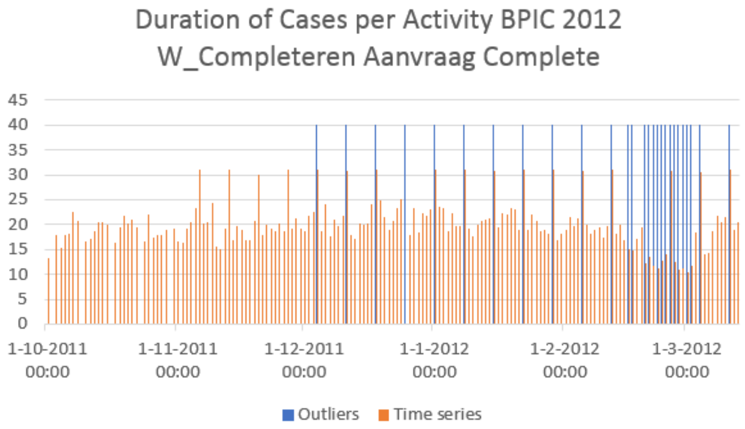


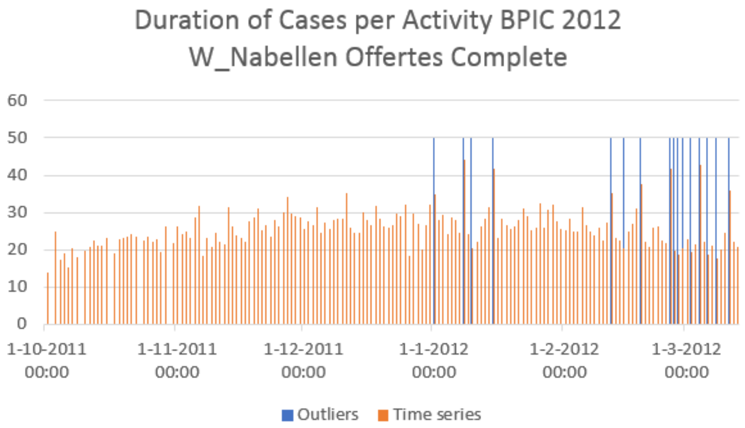
C.2 Clear peaks

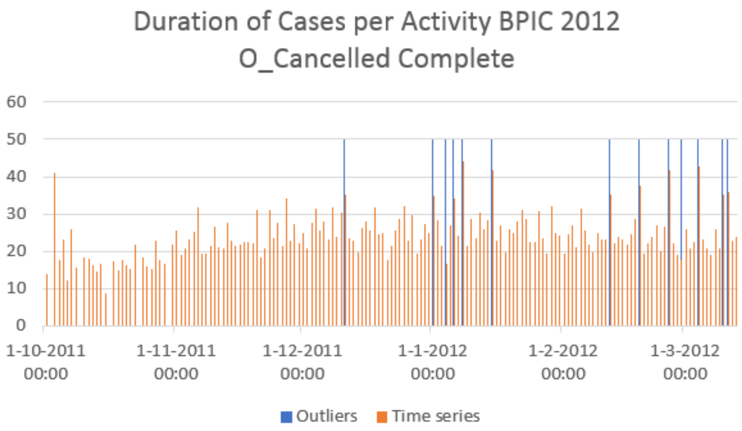


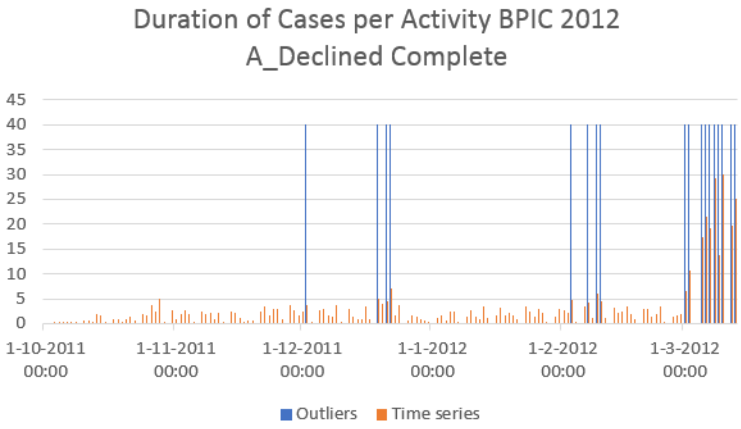


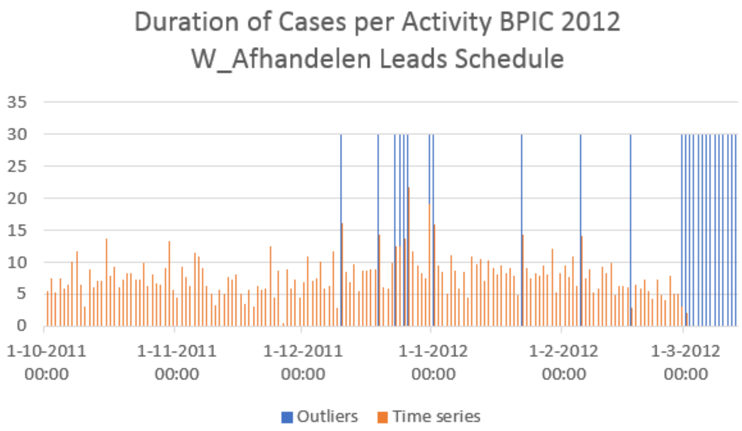

Duration of Cases per Activity BPIC 2012
A_Preaccepted Complete
35
30
25
20
15
10
5
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series

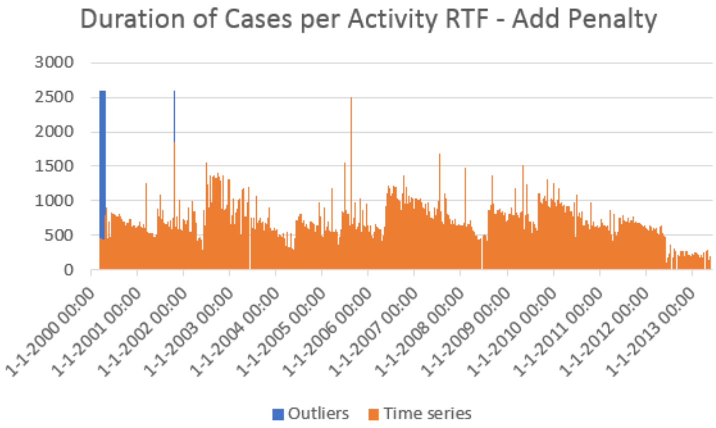

Duration of Cases per Activity BPIC 2012
W_Completeren Aanvraag Complete
45
40
35
30
25
20
15
10
5
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series

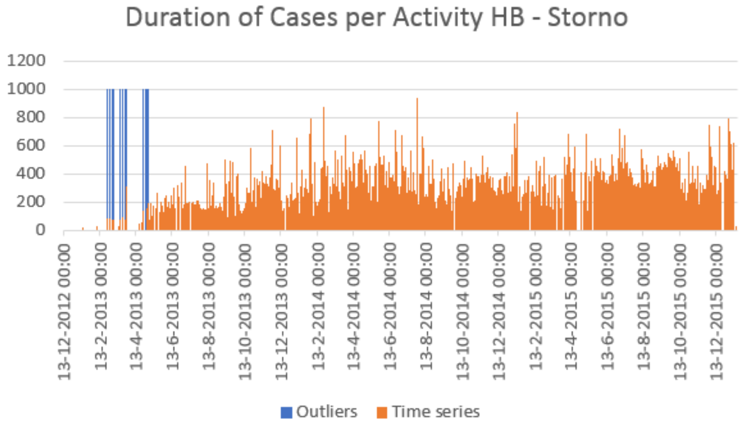

Duration of Cases per Activity BPIC 2012
W_Nabellen Offertes Complete
60
50
40
30
20
10
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series

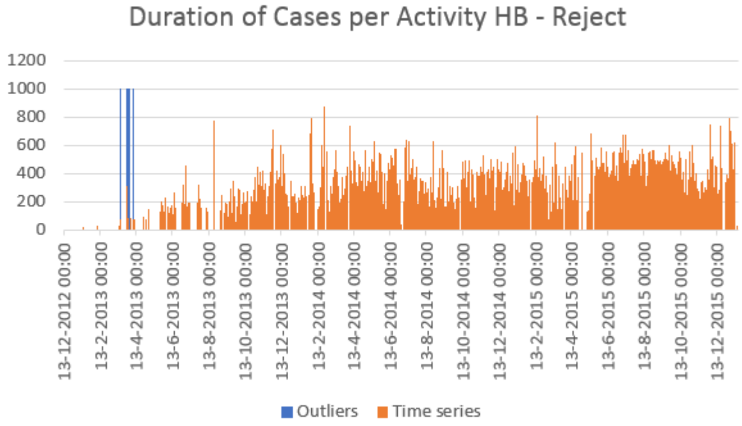

Duration of Cases per Activity BPIC 2012
O_Cancelled Complete

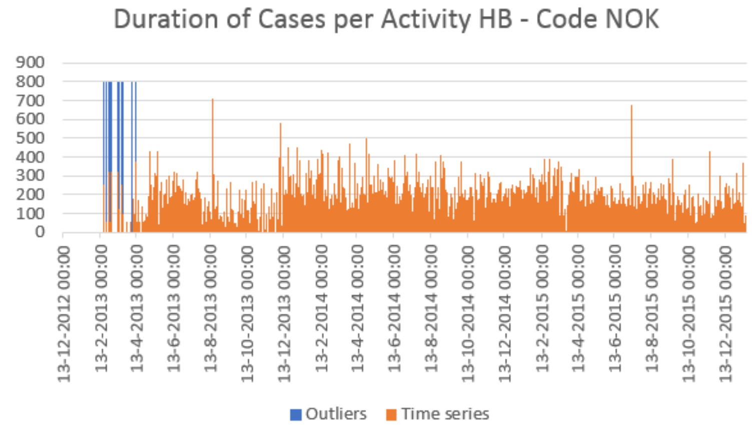

Duration of Cases per Activity BPIC 2012
A_Declined Complete

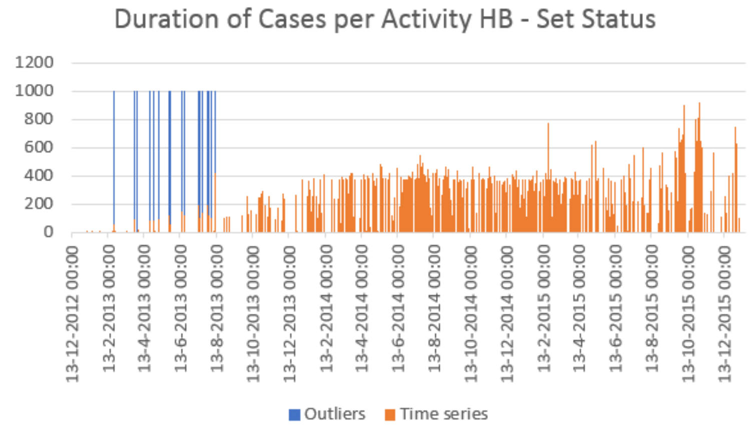

Duration of Cases per Activity BPIC 2012
W_Afhandelen Leads Schedule

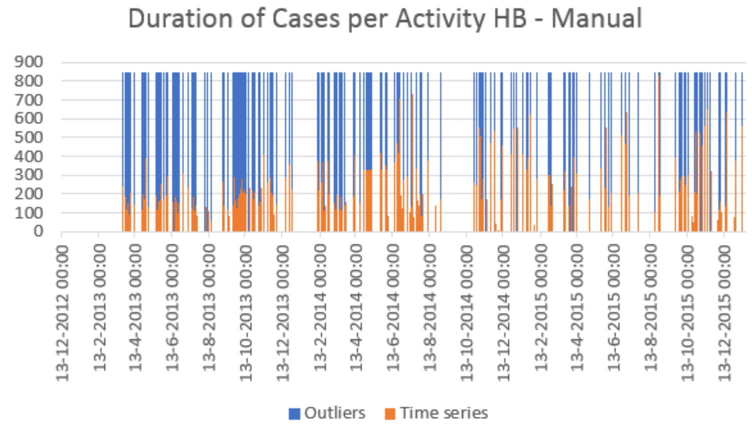

Duration of Cases per Activity RTF - Add Penalty
3000
2500
2000
1500
1000
500
0
1-1-2000 00:00
1-1-2001 00:00
1-1-2002 00:00
1-1-2003 00:00
1-1-2004 00:00
1-1-2005 00:00
1-1-2006 00:00
1-1-2007 00:00
1-1-2008 00:00
1-1-2009 00:00
1-1-2010 00:00
1-1-2011 00:00
1-1-2012 00:00
1-1-2013 00:00
🟦 Outliers
🟧 Time series

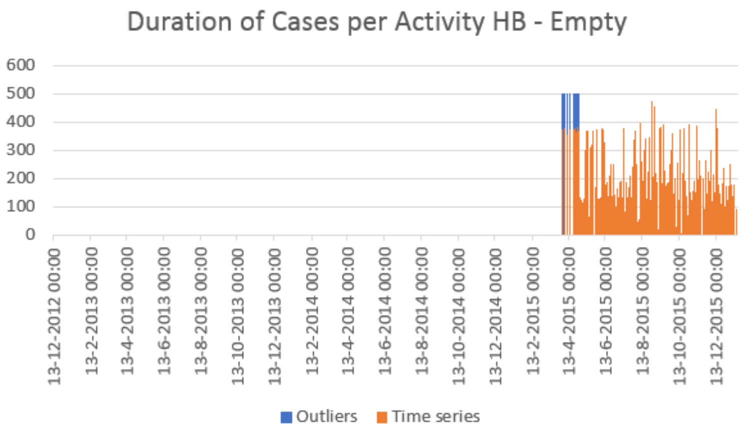

Duration of Cases per Activity HB - Storno
1200
1000
800
600
400
200
0
13-12-2012 00:00
13-2-2013 00:00
13-4-2013 00:00
13-6-2013 00:00
13-8-2013 00:00
13-10-2013 00:00
13-12-2013 00:00
13-2-2014 00:00
13-4-2014 00:00
13-6-2014 00:00
13-8-2014 00:00
13-10-2014 00:00
13-12-2014 00:00
13-2-2015 00:00
13-4-2015 00:00
13-6-2015 00:00
13-8-2015 00:00
13-10-2015 00:00
13-12-2015 00:00
🟦 Outliers
🟧 Time series


Duration of Cases per Activity HB - Reject
1200
1000
800
600
400
200
0
13-12-2012 00:00
13-2-2013 00:00
13-4-2013 00:00
13-6-2013 00:00
13-8-2013 00:00
13-10-2013 00:00
13-12-2013 00:00
13-2-2014 00:00
13-4-2014 00:00
13-6-2014 00:00
13-8-2014 00:00
13-10-2014 00:00
13-12-2014 00:00
13-2-2015 00:00
13-4-2015 00:00
13-6-2015 00:00
13-8-2015 00:00
13-10-2015 00:00
13-12-2015 00:00
🟦 Outliers
🟧 Time series

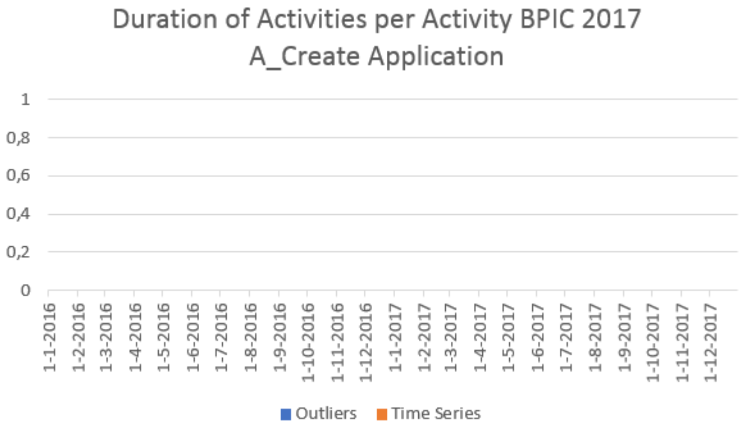

Duration of Cases per Activity HB - Code NOK
900
800
700
600
500
400
300
200
100
0
13-12-2012 00:00
13-2-2013 00:00
13-4-2013 00:00
13-6-2013 00:00
13-8-2013 00:00
13-10-2013 00:00
13-12-2013 00:00
13-2-2014 00:00
13-4-2014 00:00
13-6-2014 00:00
13-8-2014 00:00
13-10-2014 00:00
13-12-2014 00:00
13-2-2015 00:00
13-4-2015 00:00
13-6-2015 00:00
13-8-2015 00:00
13-10-2015 00:00
13-12-2015 00:00
Outliers
Time series

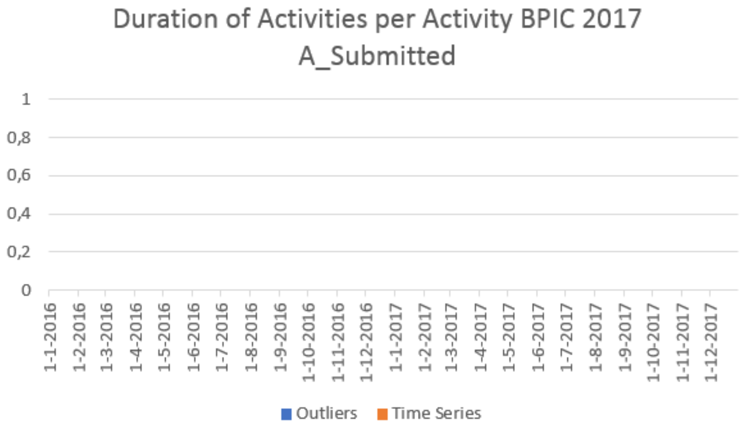

Duration of Cases per Activity HB - Set Status
1200
1000
800
600
400
200
0
13-12-2012 00:00
13-2-2013 00:00
13-4-2013 00:00
13-6-2013 00:00
13-8-2013 00:00
13-10-2013 00:00
13-12-2013 00:00
13-2-2014 00:00
13-4-2014 00:00
13-6-2014 00:00
13-8-2014 00:00
13-10-2014 00:00
13-12-2014 00:00
13-2-2015 00:00
13-4-2015 00:00
13-6-2015 00:00
13-8-2015 00:00
13-10-2015 00:00
13-12-2015 00:00
Outliers
Time series

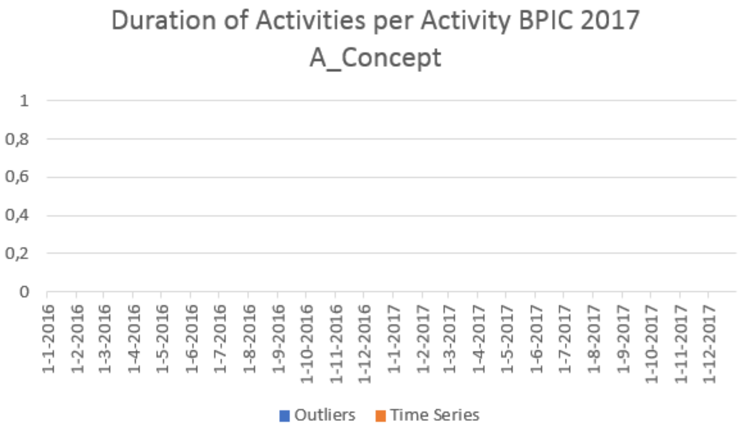

Duration of Cases per Activity HB - Manual
900
800
700
600
500
400
300
200
100
0
13-12-2012 00:00
13-2-2013 00:00
13-4-2013 00:00
13-6-2013 00:00
13-8-2013 00:00
13-10-2013 00:00
13-12-2013 00:00
13-2-2014 00:00
13-4-2014 00:00
13-6-2014 00:00
13-8-2014 00:00
13-10-2014 00:00
13-12-2014 00:00
13-2-2015 00:00
13-4-2015 00:00
13-6-2015 00:00
13-8-2015 00:00
13-10-2015 00:00
13-12-2015 00:00
Outliers
Time series

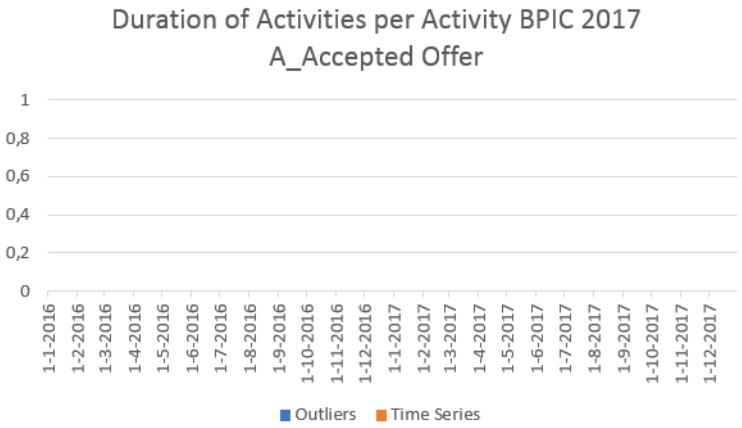


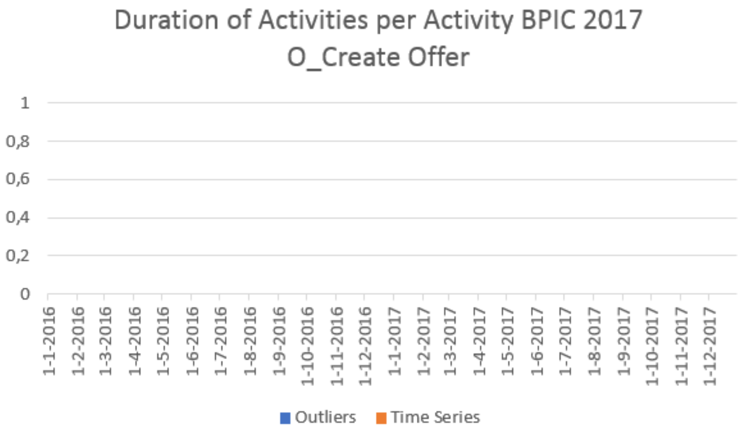
C.3 Few Values

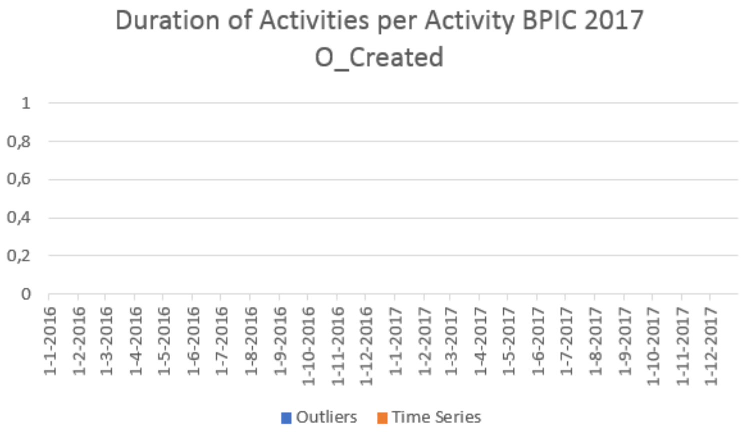


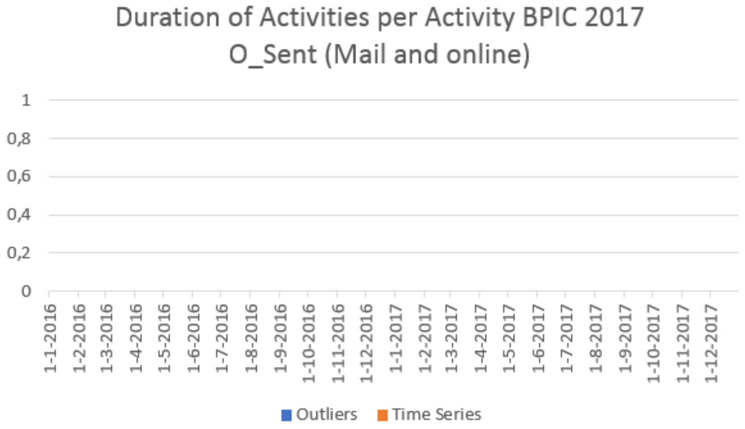


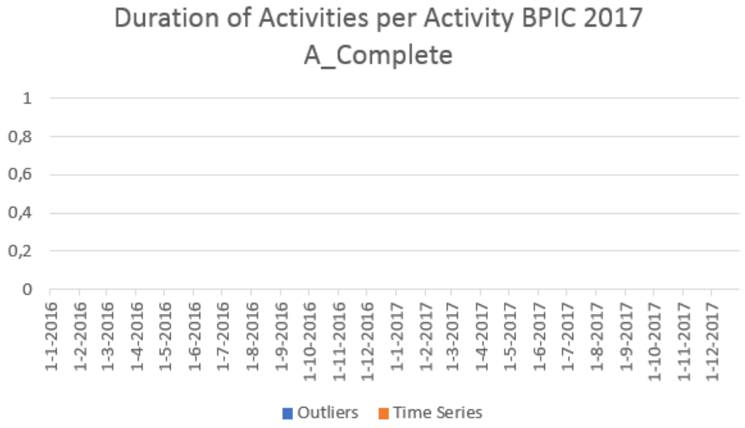

Duration of Activities per Activity BPIC 2017
A_Concept
1
0,8
0,6
0,4
0,2
0
1-1-2016
1-2-2016
1-3-2016
1-4-2016
1-5-2016
1-6-2016
1-7-2016
1-8-2016
1-9-2016
1-10-2016
1-11-2016
1-12-2016
1-1-2017
1-2-2017
1-3-2017
1-4-2017
1-5-2017
1-6-2017
1-7-2017
1-8-2017
1-9-2017
1-10-2017
1-11-2017
1-12-2017
🟦 Outliers
🟧 Time Series

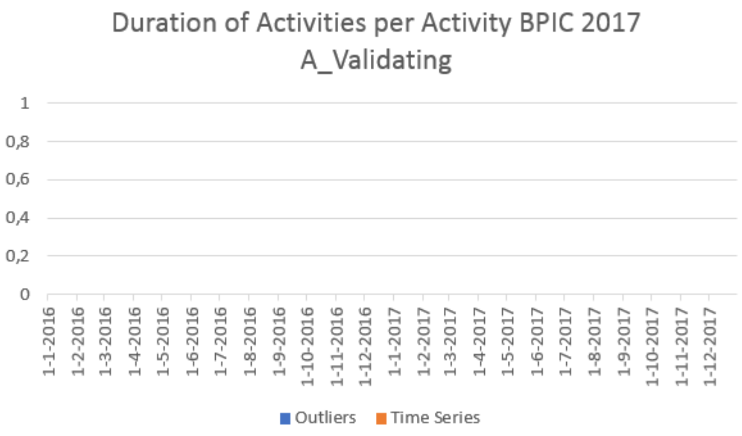

Duration of Activities per Activity BPIC 2017
A_Accepted Offer
1
0,8
0,6
0,4
0,2
0
1-1-2016
1-2-2016
1-3-2016
1-4-2016
1-5-2016
1-6-2016
1-7-2016
1-8-2016
1-9-2016
1-10-2016
1-11-2016
1-12-2016
1-1-2017
1-2-2017
1-3-2017
1-4-2017
1-5-2017
1-6-2017
1-7-2017
1-8-2017
1-9-2017
1-10-2017
1-11-2017
1-12-2017
🟦 Outliers
🟧 Time Series

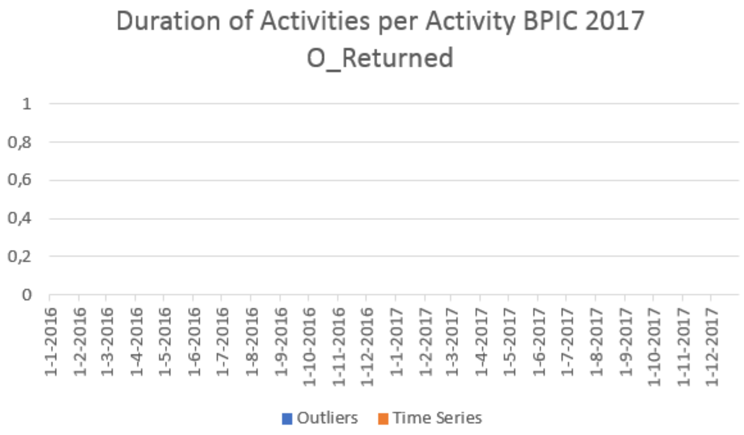

Duration of Activities per Activity BPIC 2017
O_Create Offer
1
0,8
0,6
0,4
0,2
0
1-1-2016
1-2-2016
1-3-2016
1-4-2016
1-5-2016
1-6-2016
1-7-2016
1-8-2016
1-9-2016
1-10-2016
1-11-2016
1-12-2016
1-1-2017
1-2-2017
1-3-2017
1-4-2017
1-5-2017
1-6-2017
1-7-2017
1-8-2017
1-9-2017
1-10-2017
1-11-2017
1-12-2017
🟦 Outliers
🟧 Time Series

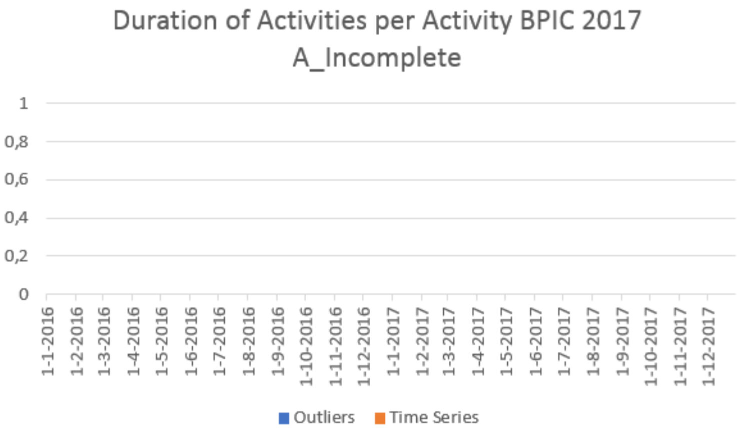

Duration of Activities per Activity BPIC 2017
O_Created
1
0,8
0,6
0,4
0,2
0
1-1-2016
1-2-2016
1-3-2016
1-4-2016
1-5-2016
1-6-2016
1-7-2016
1-8-2016
1-9-2016
1-10-2016
1-11-2016
1-12-2016
1-1-2017
1-2-2017
1-3-2017
1-4-2017
1-5-2017
1-6-2017
1-7-2017
1-8-2017
1-9-2017
1-10-2017
1-11-2017
1-12-2017
Outliers
Time Series

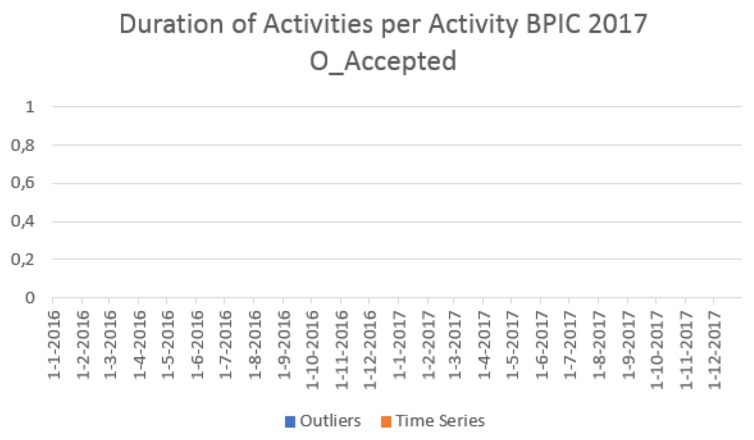

Duration of Activities per Activity BPIC 2017
O_Sent (Mail and online)
1
0,8
0,6
0,4
0,2
0
1-1-2016
1-2-2016
1-3-2016
1-4-2016
1-5-2016
1-6-2016
1-7-2016
1-8-2016
1-9-2016
1-10-2016
1-11-2016
1-12-2016
1-1-2017
1-2-2017
1-3-2017
1-4-2017
1-5-2017
1-6-2017
1-7-2017
1-8-2017
1-9-2017
1-10-2017
1-11-2017
1-12-2017
Outliers
Time Series

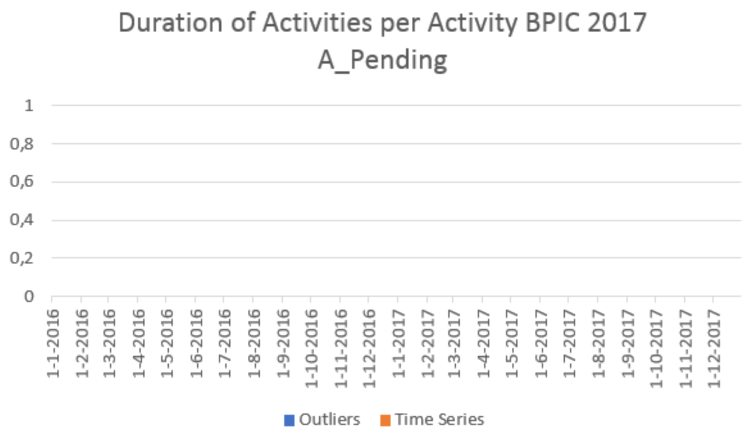

Duration of Activities per Activity BPIC 2017
A_Complete
1
0,8
0,6
0,4
0,2
0
1-1-2016
1-2-2016
1-3-2016
1-4-2016
1-5-2016
1-6-2016
1-7-2016
1-8-2016
1-9-2016
1-10-2016
1-11-2016
1-12-2016
1-1-2017
1-2-2017
1-3-2017
1-4-2017
1-5-2017
1-6-2017
1-7-2017
1-8-2017
1-9-2017
1-10-2017
1-11-2017
1-12-2017
Outliers
Time Series

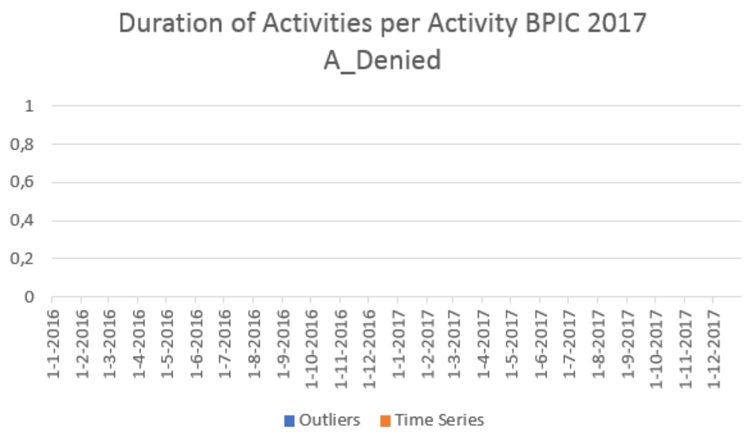

Duration of Activities per Activity BPIC 2017
A_Validating
1
0,8
0,6
0,4
0,2
0
1-1-2016
1-2-2016
1-3-2016
1-4-2016
1-5-2016
1-6-2016
1-7-2016
1-8-2016
1-9-2016
1-10-2016
1-11-2016
1-12-2016
1-1-2017
1-2-2017
1-3-2017
1-4-2017
1-5-2017
1-6-2017
1-7-2017
1-8-2017
1-9-2017
1-10-2017
1-11-2017
1-12-2017
🟦 Outliers
🟧 Time Series

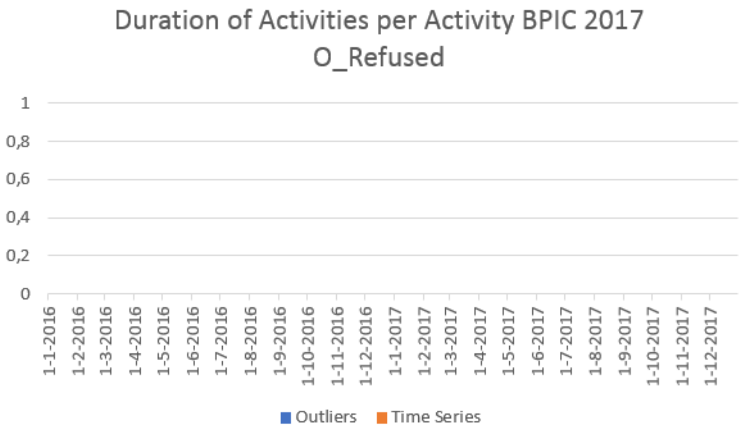

Duration of Activities per Activity BPIC 2017
O_Returned
1
0,8
0,6
0,4
0,2
0
1-1-2016
1-2-2016
1-3-2016
1-4-2016
1-5-2016
1-6-2016
1-7-2016
1-8-2016
1-9-2016
1-10-2016
1-11-2016
1-12-2016
1-1-2017
1-2-2017
1-3-2017
1-4-2017
1-5-2017
1-6-2017
1-7-2017
1-8-2017
1-9-2017
1-10-2017
1-11-2017
1-12-2017
🟦 Outliers
🟧 Time Series

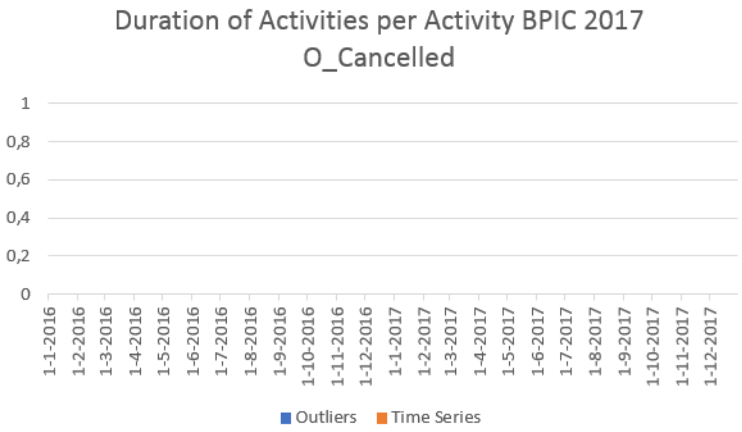

Duration of Activities per Activity BPIC 2017
A_Incomplete
1
0,8
0,6
0,4
0,2
0
1-1-2016
1-2-2016
1-3-2016
1-4-2016
1-5-2016
1-6-2016
1-7-2016
1-8-2016
1-9-2016
1-10-2016
1-11-2016
1-12-2016
1-1-2017
1-2-2017
1-3-2017
1-4-2017
1-5-2017
1-6-2017
1-7-2017
1-8-2017
1-9-2017
1-10-2017
1-11-2017
1-12-2017
🟦 Outliers
🟧 Time Series

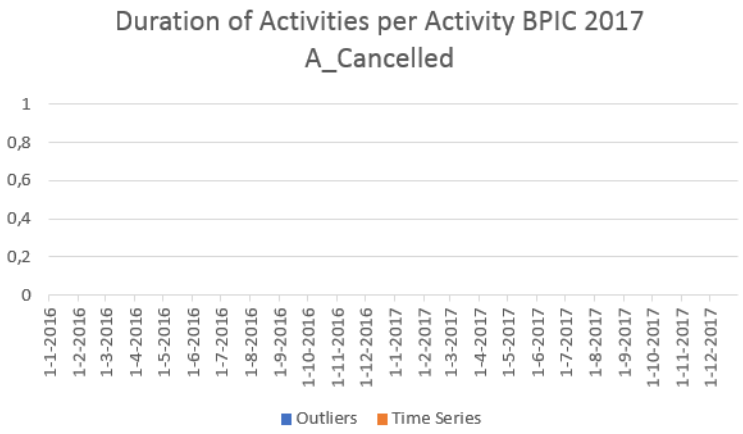

Duration of Activities per Activity BPIC 2017
O_Accepted
1
0,8
0,6
0,4
0,2
0
1-1-2016
1-2-2016
1-3-2016
1-4-2016
1-5-2016
1-6-2016
1-7-2016
1-8-2016
1-9-2016
1-10-2016
1-11-2016
1-12-2016
1-1-2017
1-2-2017
1-3-2017
1-4-2017
1-5-2017
1-6-2017
1-7-2017
1-8-2017
1-9-2017
1-10-2017
1-11-2017
1-12-2017
■ Outliers ■ Time Series

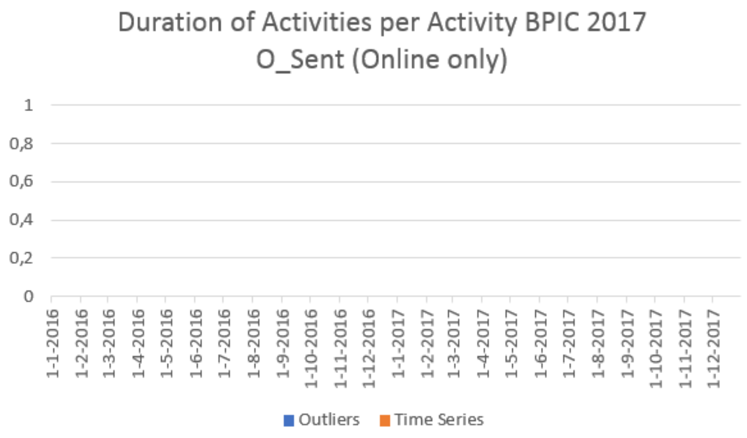

Duration of Activities per Activity BPIC 2017
A_Pending
1
0,8
0,6
0,4
0,2
0
1-1-2016
1-2-2016
1-3-2016
1-4-2016
1-5-2016
1-6-2016
1-7-2016
1-8-2016
1-9-2016
1-10-2016
1-11-2016
1-12-2016
1-1-2017
1-2-2017
1-3-2017
1-4-2017
1-5-2017
1-6-2017
1-7-2017
1-8-2017
1-9-2017
1-10-2017
1-11-2017
1-12-2017
■ Outliers ■ Time Series


Duration of Activities per Activity BPIC 2017
A_Denied
1
0,8
0,6
0,4
0,2
0
1-1-2016
1-2-2016
1-3-2016
1-4-2016
1-5-2016
1-6-2016
1-7-2016
1-8-2016
1-9-2016
1-10-2016
1-11-2016
1-12-2016
1-1-2017
1-2-2017
1-3-2017
1-4-2017
1-5-2017
1-6-2017
1-7-2017
1-8-2017
1-9-2017
1-10-2017
1-11-2017
1-12-2017
■ Outliers ■ Time Series

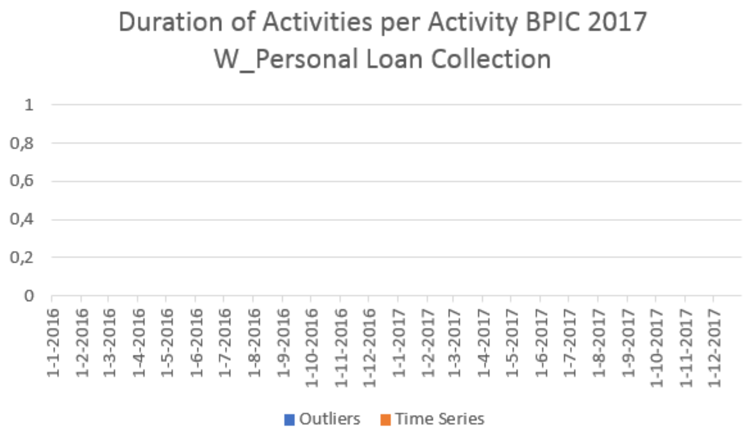

Duration of Activities per Activity BPIC 2017
O_Refused
1
0,8
0,6
0,4
0,2
0
1-1-2016
1-2-2016
1-3-2016
1-4-2016
1-5-2016
1-6-2016
1-7-2016
1-8-2016
1-9-2016
1-10-2016
1-11-2016
1-12-2016
1-1-2017
1-2-2017
1-3-2017
1-4-2017
1-5-2017
1-6-2017
1-7-2017
1-8-2017
1-9-2017
1-10-2017
1-11-2017
1-12-2017
Outliers
Time Series


Duration of Activities per Activity BPIC 2017
O_Cancelled
1
0,8
0,6
0,4
0,2
0
1-1-2016
1-2-2016
1-3-2016
1-4-2016
1-5-2016
1-6-2016
1-7-2016
1-8-2016
1-9-2016
1-10-2016
1-11-2016
1-12-2016
1-1-2017
1-2-2017
1-3-2017
1-4-2017
1-5-2017
1-6-2017
1-7-2017
1-8-2017
1-9-2017
1-10-2017
1-11-2017
1-12-2017
Outliers
Time Series

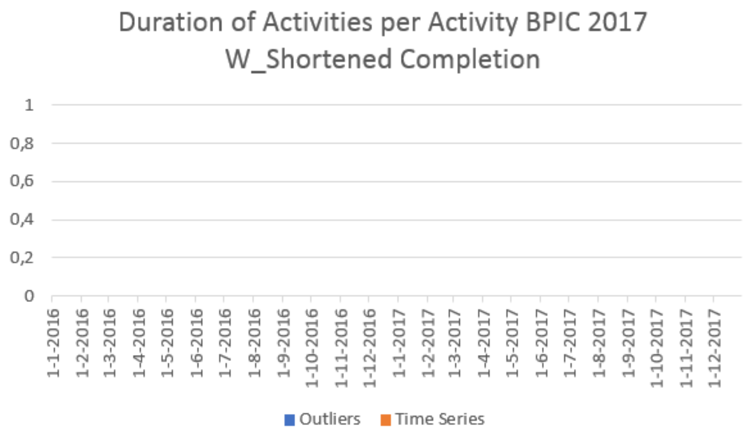

Duration of Activities per Activity BPIC 2017
A_Cancelled
1
0,8
0,6
0,4
0,2
0
1-1-2016
1-2-2016
1-3-2016
1-4-2016
1-5-2016
1-6-2016
1-7-2016
1-8-2016
1-9-2016
1-10-2016
1-11-2016
1-12-2016
1-1-2017
1-2-2017
1-3-2017
1-4-2017
1-5-2017
1-6-2017
1-7-2017
1-8-2017
1-9-2017
1-10-2017
1-11-2017
1-12-2017
Outliers
Time Series



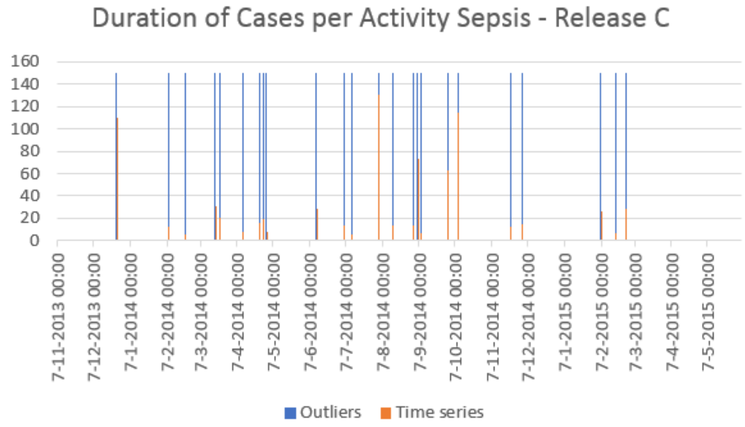
Duration of Activities per Activity BPIC 2017
O_Sent (Online only)

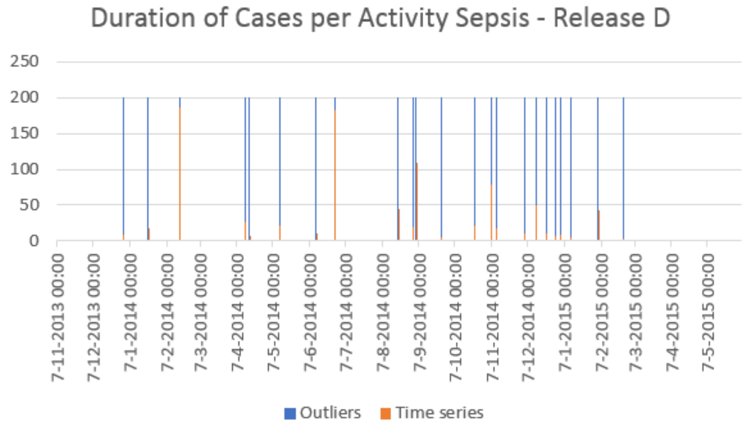


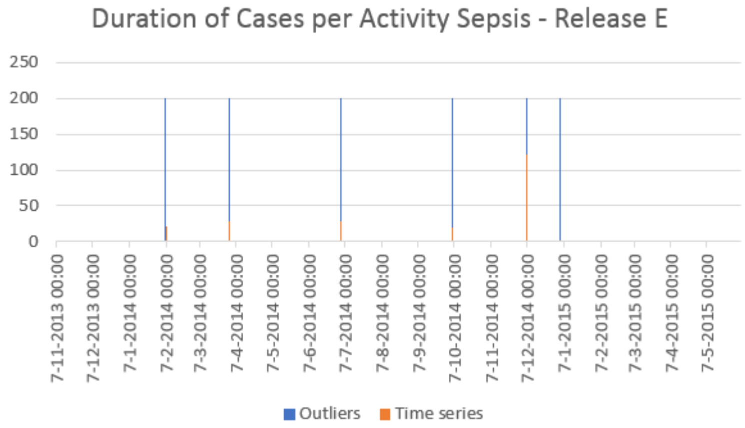
Duration of Activities per Activity BPIC 2017
W_Personal Loan Collection

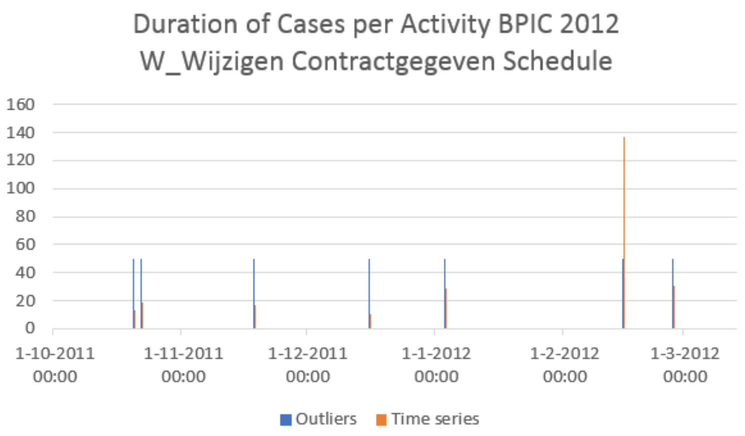


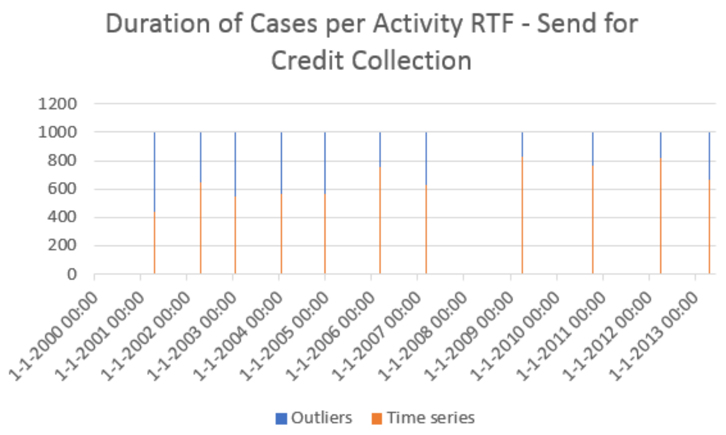
Duration of Activities per Activity BPIC 2017
W_Shortened Completion

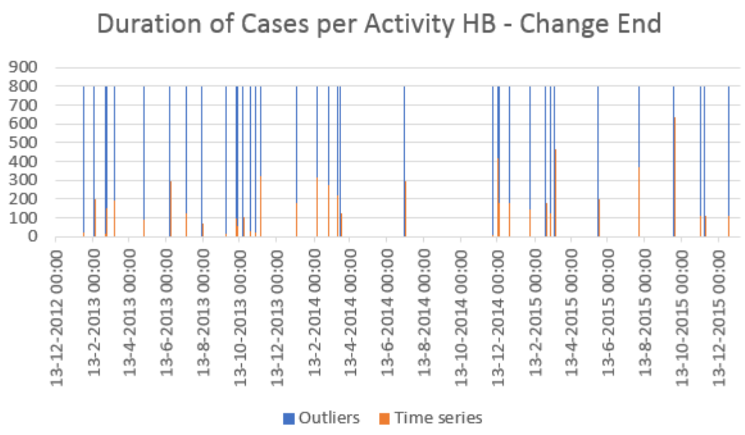

Duration of Cases per Activity Sepsis - Release C

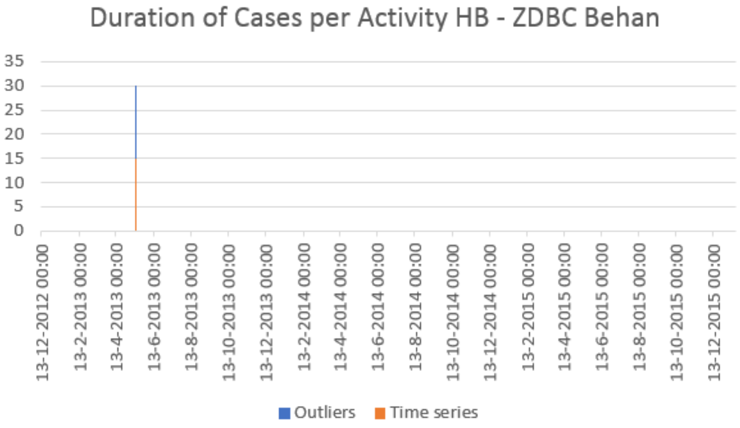

Duration of Cases per Activity Sepsis - Release D


Duration of Cases per Activity Sepsis - Release E

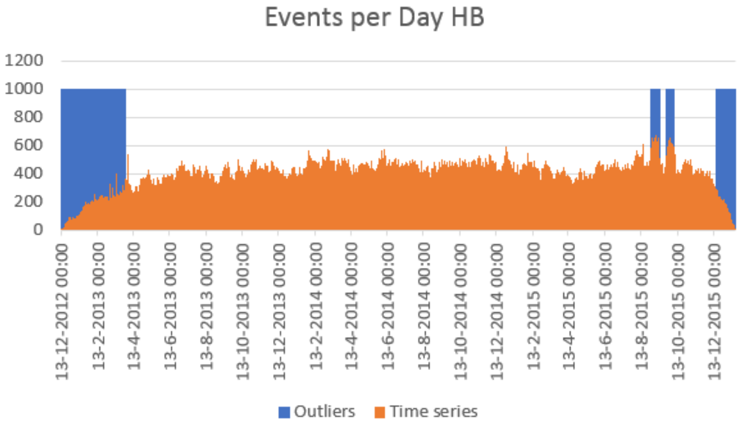

Duration of Cases per Activity BPIC 2012
W_Wijzigen Contractgegeven Schedule
160
140
120
100
80
60
40
20
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series

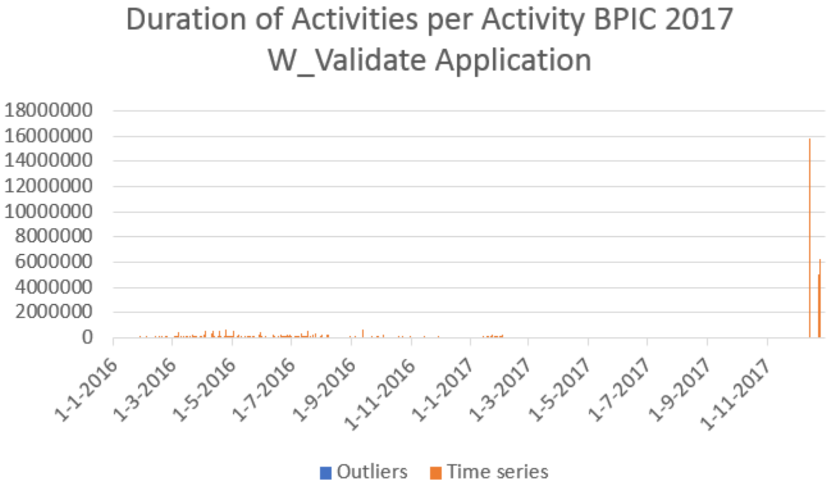

Duration of Cases per Activity RTF - Send for Credit Collection
1200
1000
800
600
400
200
0
1-1-2000 00:00
1-1-2001 00:00
1-1-2002 00:00
1-1-2003 00:00
1-1-2004 00:00
1-1-2005 00:00
1-1-2006 00:00
1-1-2007 00:00
1-1-2008 00:00
1-1-2009 00:00
1-1-2010 00:00
1-1-2011 00:00
1-1-2012 00:00
1-1-2013 00:00
Outliers
Time series

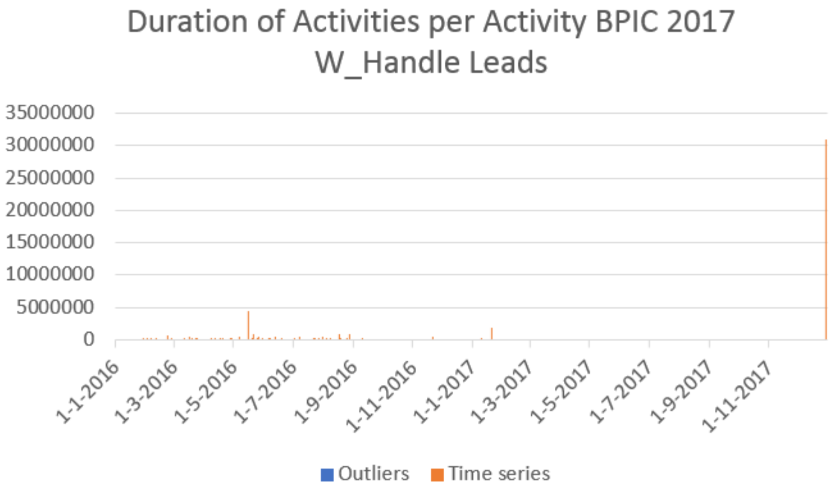

Duration of Cases per Activity HB - Change End
900
800
700
600
500
400
300
200
100
0
13-12-2012 00:00
13-2-2013 00:00
13-4-2013 00:00
13-6-2013 00:00
13-8-2013 00:00
13-10-2013 00:00
13-12-2013 00:00
13-2-2014 00:00
13-4-2014 00:00
13-6-2014 00:00
13-8-2014 00:00
13-10-2014 00:00
13-12-2014 00:00
13-2-2015 00:00
13-4-2015 00:00
13-6-2015 00:00
13-8-2015 00:00
13-10-2015 00:00
13-12-2015 00:00
Outliers
Time series

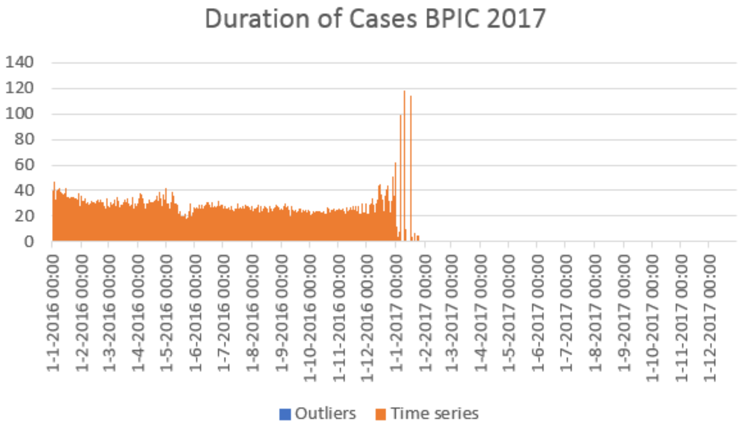


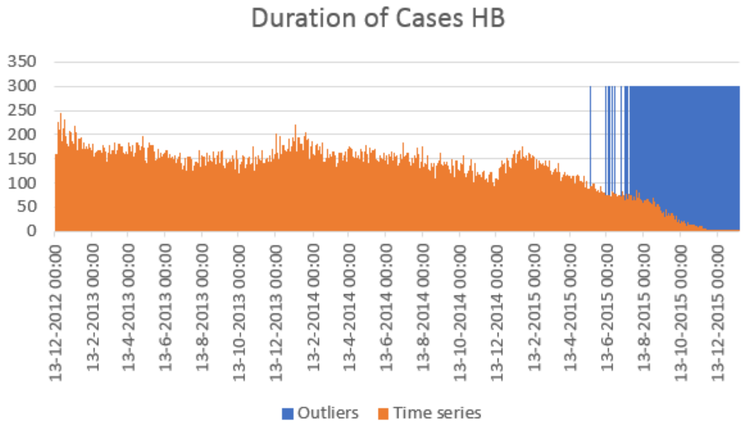
C.4 Smooth time series

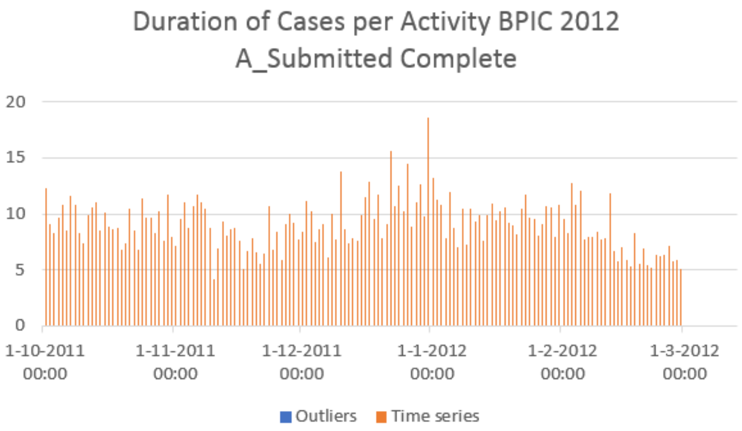


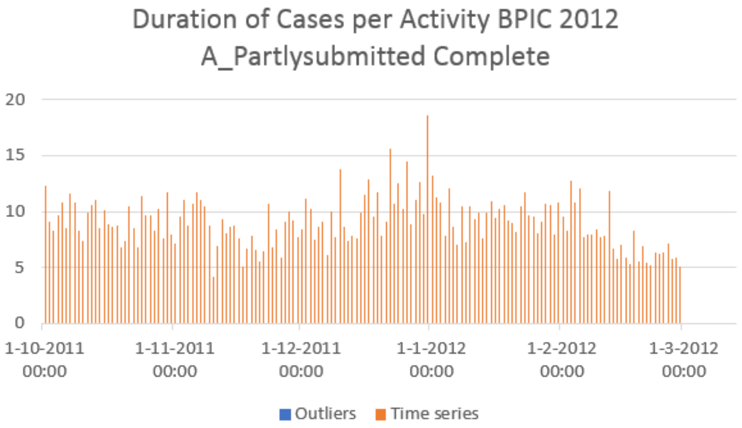


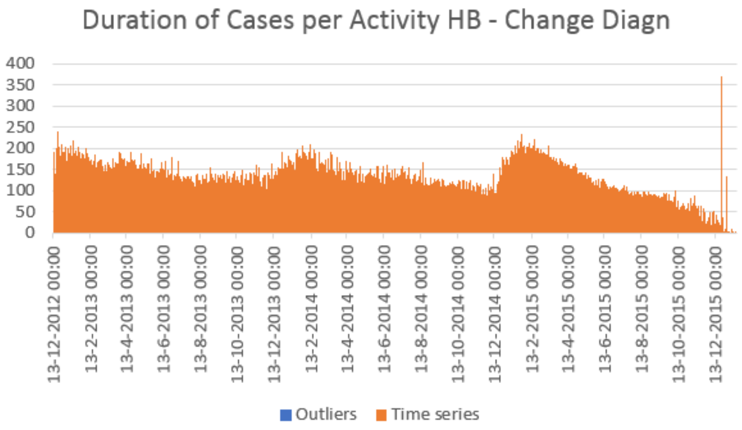

Duration of Activities per Activity BPIC 2017 W_Handle Leads


Duration of Cases BPIC 2017

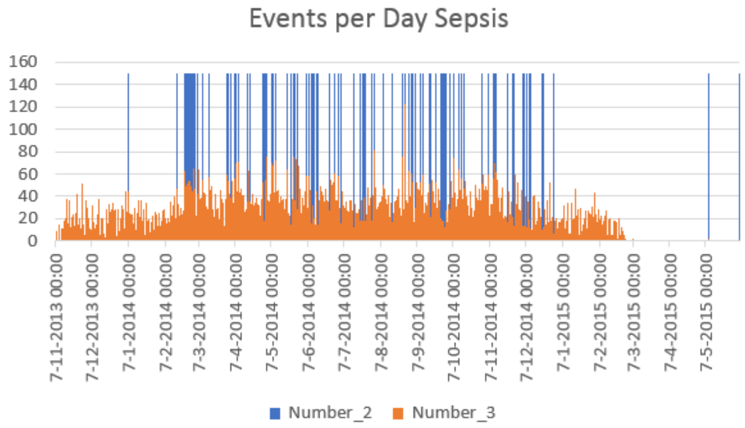

Duration of Cases HB

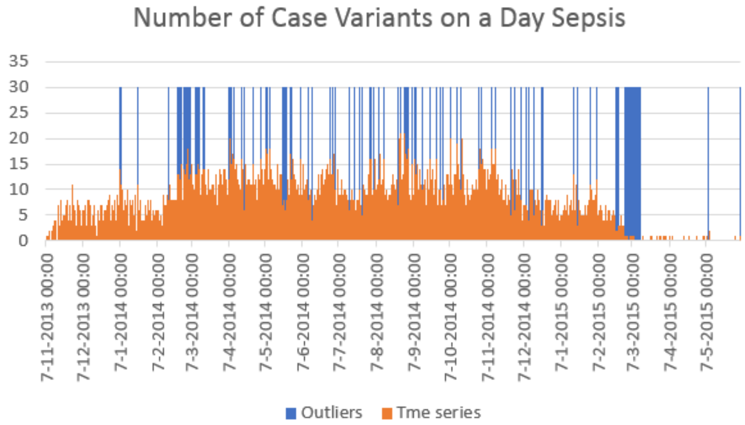

Duration of Cases per Activity BPIC 2012
A_Submitted Complete
20
15
10
5
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series

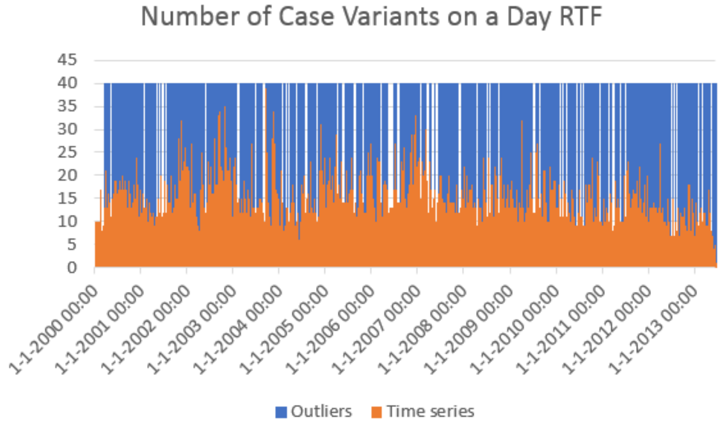

Duration of Cases per Activity BPIC 2012
A_Partlysubmitted Complete
20
15
10
5
0
1-10-2011 00:00
1-11-2011 00:00
1-12-2011 00:00
1-1-2012 00:00
1-2-2012 00:00
1-3-2012 00:00
Outliers
Time series

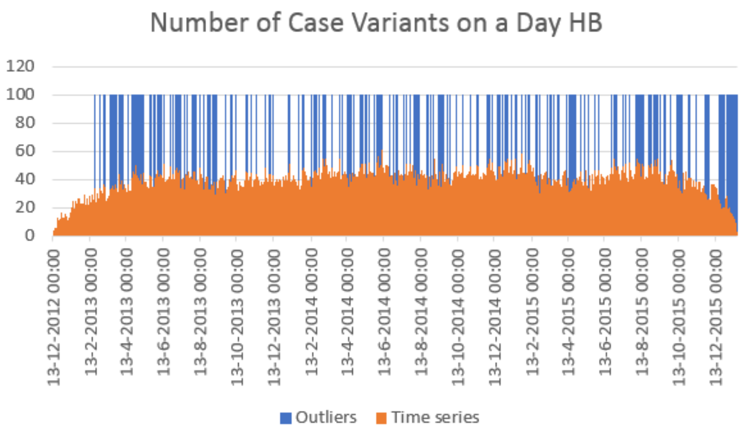

Duration of Cases per Activity HB - Change Diagn
400
350
300
250
200
150
100
50
0
13-12-2012 00:00
13-2-2013 00:00
13-4-2013 00:00
13-6-2013 00:00
13-8-2013 00:00
13-10-2013 00:00
13-12-2013 00:00
13-2-2014 00:00
13-4-2014 00:00
13-6-2014 00:00
13-8-2014 00:00
13-10-2014 00:00
13-12-2014 00:00
13-2-2015 00:00
13-4-2015 00:00
13-6-2015 00:00
13-8-2015 00:00
13-10-2015 00:00
13-12-2015 00:00
Outliers
Time series

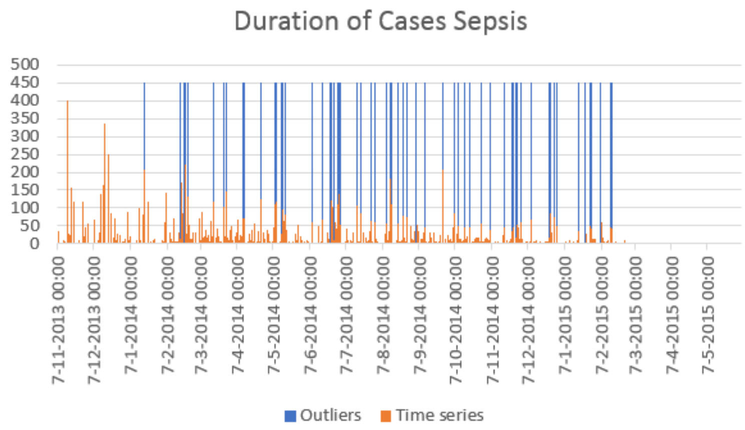
C.5 Random time series

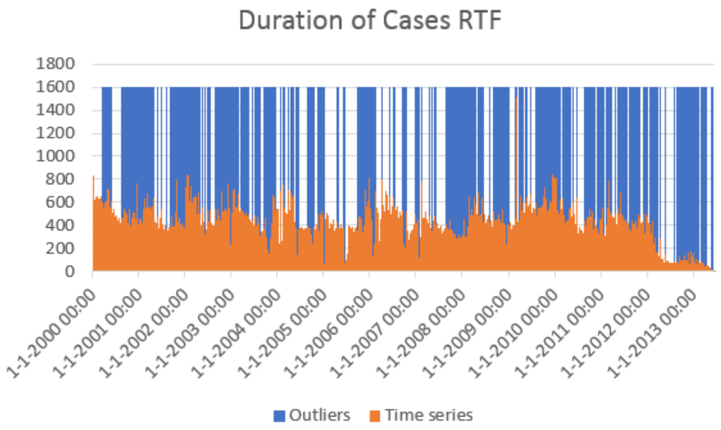


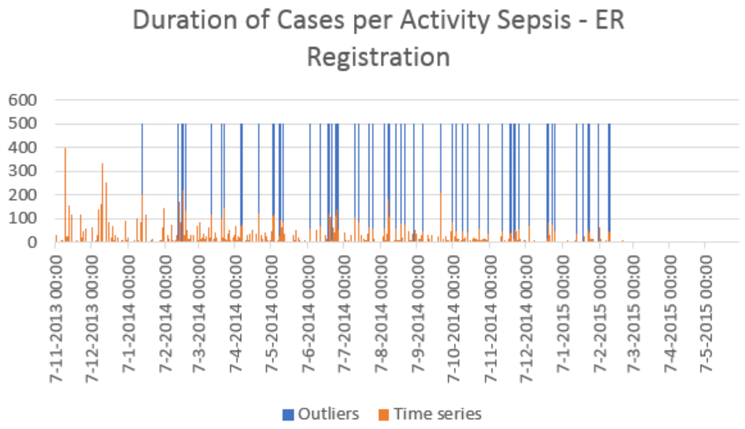


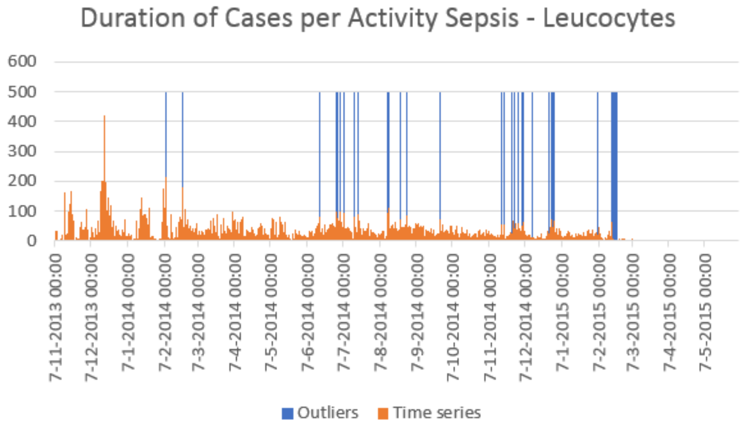


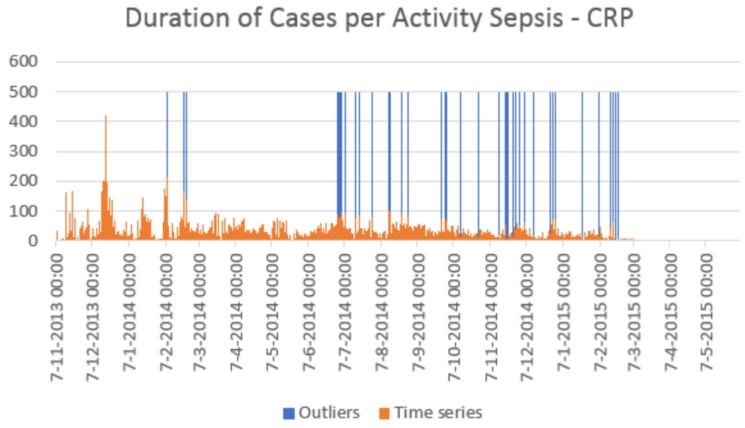

Number of Case Variants on a Day HB
120
100
80
60
40
20
0
13-12-2012 00:00
13-2-2013 00:00
13-4-2013 00:00
13-6-2013 00:00
13-8-2013 00:00
13-10-2013 00:00
13-12-2013 00:00
13-2-2014 00:00
13-4-2014 00:00
13-6-2014 00:00
13-8-2014 00:00
13-10-2014 00:00
13-12-2014 00:00
13-2-2015 00:00
13-4-2015 00:00
13-6-2015 00:00
13-8-2015 00:00
13-10-2015 00:00
13-12-2015 00:00
Outliers
Time series

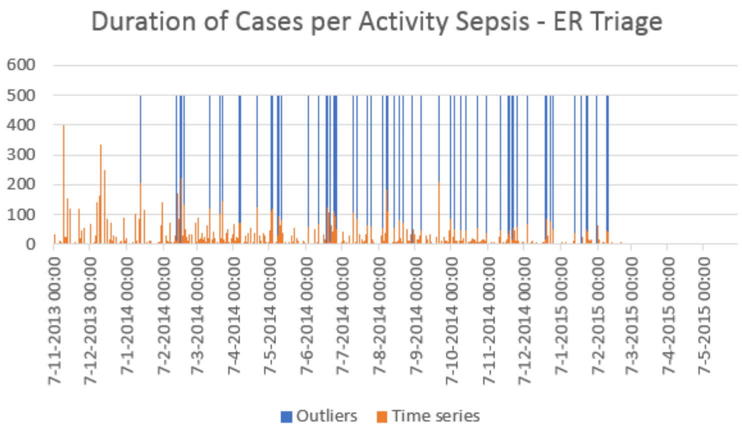

Duration of Cases Sepsis
500
450
400
350
300
250
200
150
100
50
0
7-11-2013 00:00
7-12-2013 00:00
7-1-2014 00:00
7-2-2014 00:00
7-3-2014 00:00
7-4-2014 00:00
7-5-2014 00:00
7-6-2014 00:00
7-7-2014 00:00
7-8-2014 00:00
7-9-2014 00:00
7-10-2014 00:00
7-11-2014 00:00
7-12-2014 00:00
7-1-2015 00:00
7-2-2015 00:00
7-3-2015 00:00
7-4-2015 00:00
7-5-2015 00:00
Outliers
Time series

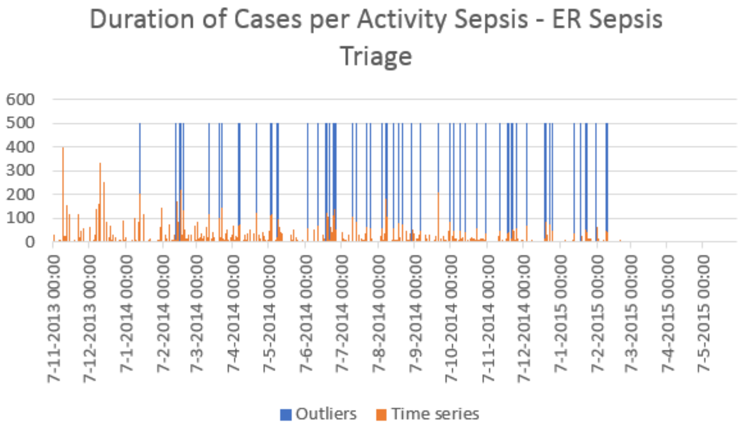

Duration of Cases RTF
1800
1600
1400
1200
1000
800
600
400
200
0
1-1-2000 00:00
1-1-2001 00:00
1-1-2002 00:00
1-1-2003 00:00
1-1-2004 00:00
1-1-2005 00:00
1-1-2006 00:00
1-1-2007 00:00
1-1-2008 00:00
1-1-2009 00:00
1-1-2010 00:00
1-1-2011 00:00
1-1-2012 00:00
1-1-2013 00:00
Outliers
Time series

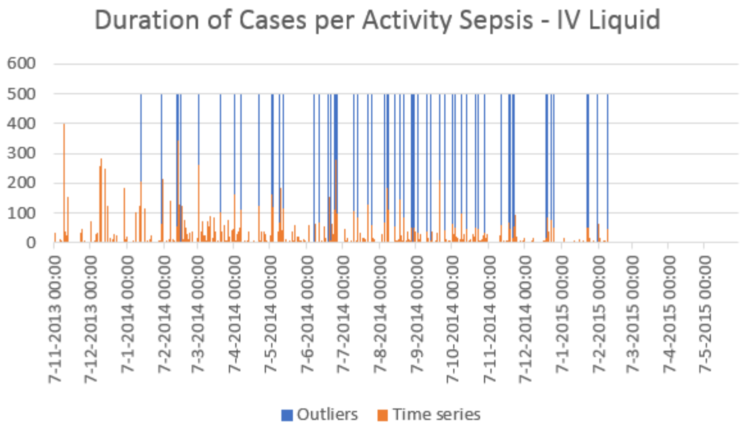

Duration of Cases per Activity Sepsis - ER Registration
600
500
400
300
200
100
0
7-11-2013 00:00
7-12-2013 00:00
7-1-2014 00:00
7-2-2014 00:00
7-3-2014 00:00
7-4-2014 00:00
7-5-2014 00:00
7-6-2014 00:00
7-7-2014 00:00
7-8-2014 00:00
7-9-2014 00:00
7-10-2014 00:00
7-11-2014 00:00
7-12-2014 00:00
7-1-2015 00:00
7-2-2015 00:00
7-3-2015 00:00
7-4-2015 00:00
7-5-2015 00:00
Outliers
Time series

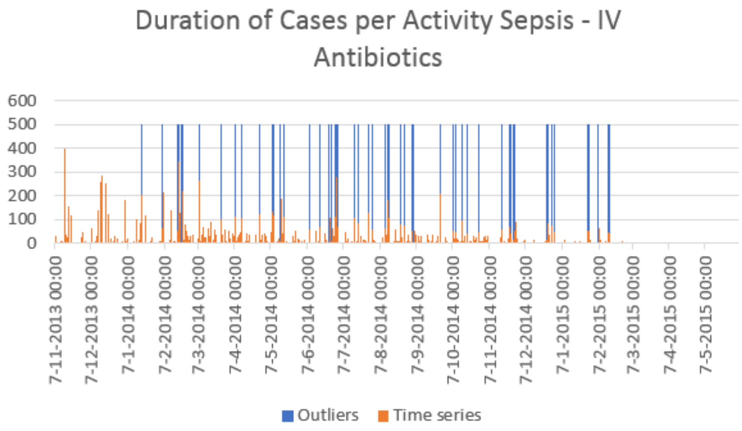

Duration of Cases per Activity Sepsis - Leucocytes
600
500
400
300
200
100
0
7-11-2013 00:00
7-12-2013 00:00
7-1-2014 00:00
7-2-2014 00:00
7-3-2014 00:00
7-4-2014 00:00
7-5-2014 00:00
7-6-2014 00:00
7-7-2014 00:00
7-8-2014 00:00
7-9-2014 00:00
7-10-2014 00:00
7-11-2014 00:00
7-12-2014 00:00
7-1-2015 00:00
7-2-2015 00:00
7-3-2015 00:00
7-4-2015 00:00
7-5-2015 00:00
Outliers
Time series

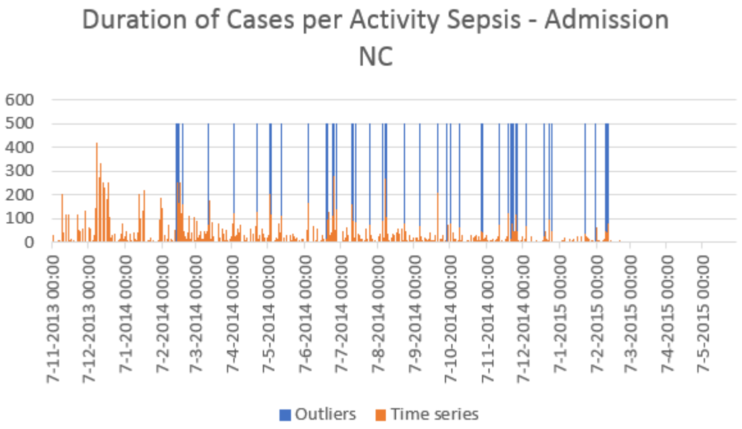

Duration of Cases per Activity Sepsis - CRP
600
500
400
300
200
100
0
7-11-2013 00:00
7-12-2013 00:00
7-1-2014 00:00
7-2-2014 00:00
7-3-2014 00:00
7-4-2014 00:00
7-5-2014 00:00
7-6-2014 00:00
7-7-2014 00:00
7-8-2014 00:00
7-9-2014 00:00
7-10-2014 00:00
7-11-2014 00:00
7-12-2014 00:00
7-1-2015 00:00
7-2-2015 00:00
7-3-2015 00:00
7-4-2015 00:00
7-5-2015 00:00
Outliers
Time series

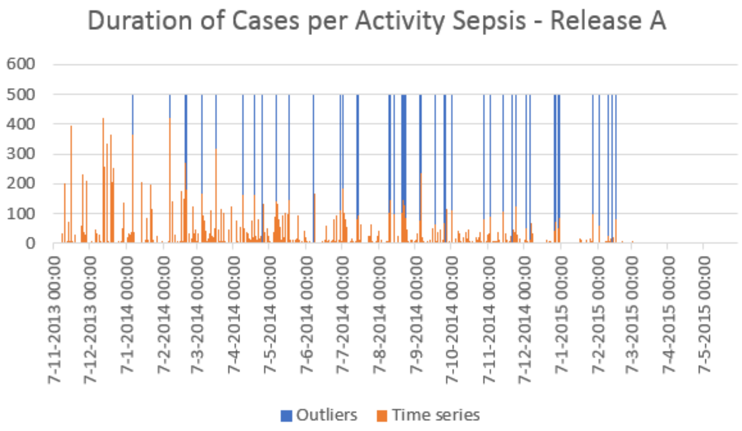

Duration of Cases per Activity Sepsis - ER Triage
600
500
400
300
200
100
0
7-11-2013 00:00
7-12-2013 00:00
7-1-2014 00:00
7-2-2014 00:00
7-3-2014 00:00
7-4-2014 00:00
7-5-2014 00:00
7-6-2014 00:00
7-7-2014 00:00
7-8-2014 00:00
7-9-2014 00:00
7-10-2014 00:00
7-11-2014 00:00
7-12-2014 00:00
7-1-2015 00:00
7-2-2015 00:00
7-3-2015 00:00
7-4-2015 00:00
7-5-2015 00:00
■ Outliers
■ Time series

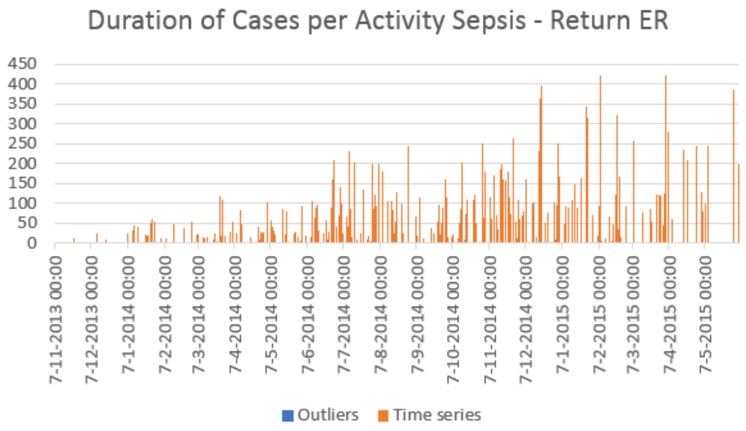

Duration of Cases per Activity Sepsis - ER Sepsis Triage
600
500
400
300
200
100
0
7-11-2013 00:00
7-12-2013 00:00
7-1-2014 00:00
7-2-2014 00:00
7-3-2014 00:00
7-4-2014 00:00
7-5-2014 00:00
7-6-2014 00:00
7-7-2014 00:00
7-8-2014 00:00
7-9-2014 00:00
7-10-2014 00:00
7-11-2014 00:00
7-12-2014 00:00
7-1-2015 00:00
7-2-2015 00:00
7-3-2015 00:00
7-4-2015 00:00
7-5-2015 00:00
■ Outliers
■ Time series

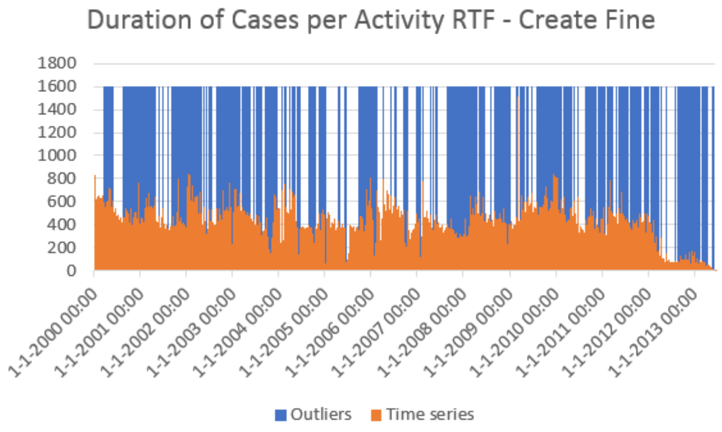

Duration of Cases per Activity Sepsis - IV Liquid
600
500
400
300
200
100
0
7-11-2013 00:00
7-12-2013 00:00
7-1-2014 00:00
7-2-2014 00:00
7-3-2014 00:00
7-4-2014 00:00
7-5-2014 00:00
7-6-2014 00:00
7-7-2014 00:00
7-8-2014 00:00
7-9-2014 00:00
7-10-2014 00:00
7-11-2014 00:00
7-12-2014 00:00
7-1-2015 00:00
7-2-2015 00:00
7-3-2015 00:00
7-4-2015 00:00
7-5-2015 00:00
■ Outliers
■ Time series

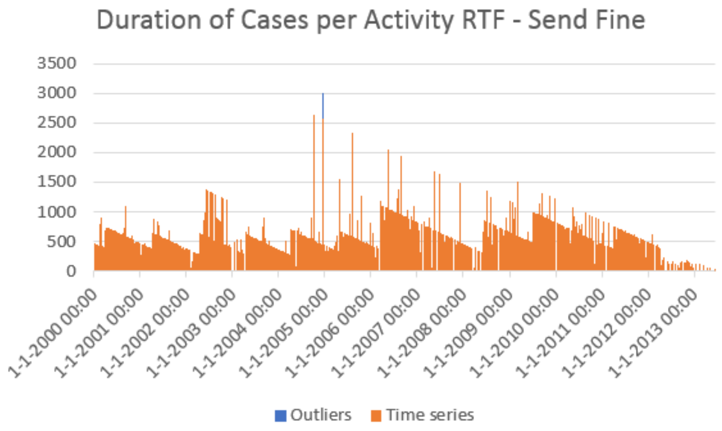

Duration of Cases per Activity Sepsis - IV Antibiotics

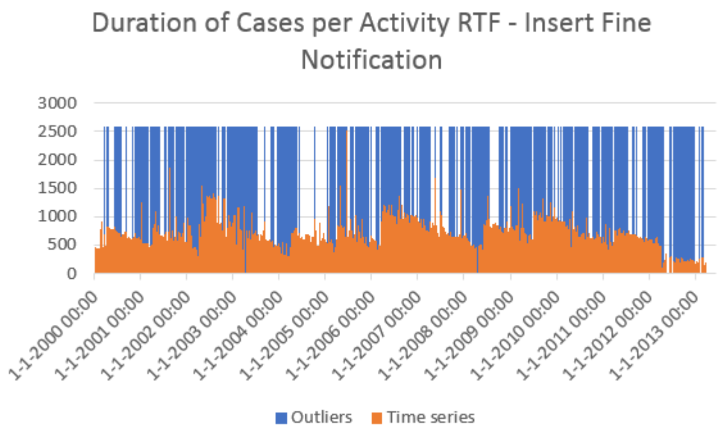

Duration of Cases per Activity Sepsis - Admission NC


Duration of Cases per Activity Sepsis - Release A

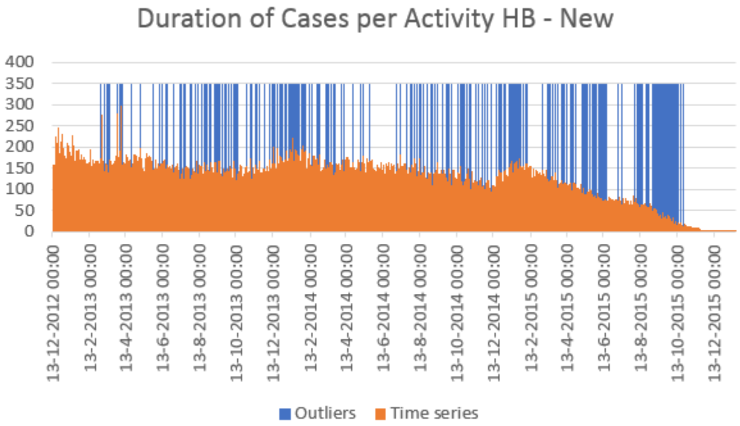

Duration of Cases per Activity Sepsis - Return ER
450
400
350
300
250
200
150
100
50
0
7-11-2013 00:00
7-12-2013 00:00
7-1-2014 00:00
7-2-2014 00:00
7-3-2014 00:00
7-4-2014 00:00
7-5-2014 00:00
7-6-2014 00:00
7-7-2014 00:00
7-8-2014 00:00
7-9-2014 00:00
7-10-2014 00:00
7-11-2014 00:00
7-12-2014 00:00
7-1-2015 00:00
7-2-2015 00:00
7-3-2015 00:00
7-4-2015 00:00
7-5-2015 00:00
Outliers
Time series


Duration of Cases per Activity RTF - Create Fine
1800
1600
1400
1200
1000
800
600
400
200
0
1-1-2000 00:00
1-1-2001 00:00
1-1-2002 00:00
1-1-2003 00:00
1-1-2004 00:00
1-1-2005 00:00
1-1-2006 00:00
1-1-2007 00:00
1-1-2008 00:00
1-1-2009 00:00
1-1-2010 00:00
1-1-2011 00:00
1-1-2012 00:00
1-1-2013 00:00
Outliers
Time series

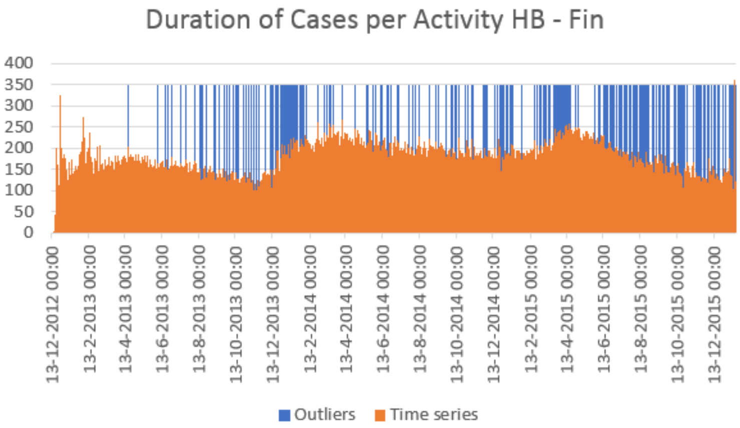

Duration of Cases per Activity RTF - Send Fine
3500
3000
2500
2000
1500
1000
500
0
1-1-2000 00:00
1-1-2001 00:00
1-1-2002 00:00
1-1-2003 00:00
1-1-2004 00:00
1-1-2005 00:00
1-1-2006 00:00
1-1-2007 00:00
1-1-2008 00:00
1-1-2009 00:00
1-1-2010 00:00
1-1-2011 00:00
1-1-2012 00:00
1-1-2013 00:00
Outliers
Time series



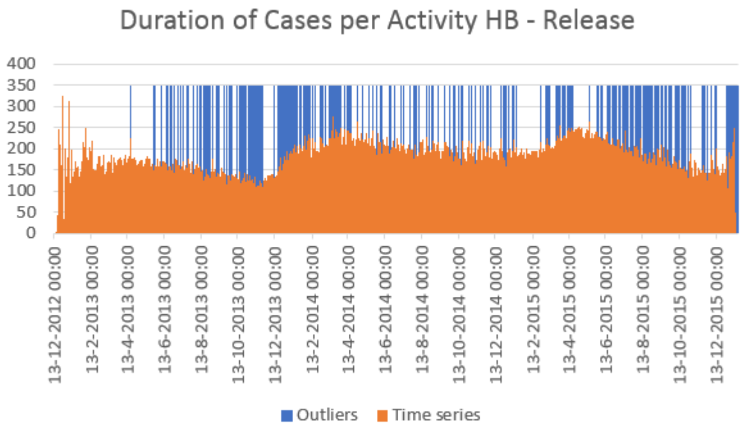
Duration of Cases per Activity RTF - Insert Fine Notification

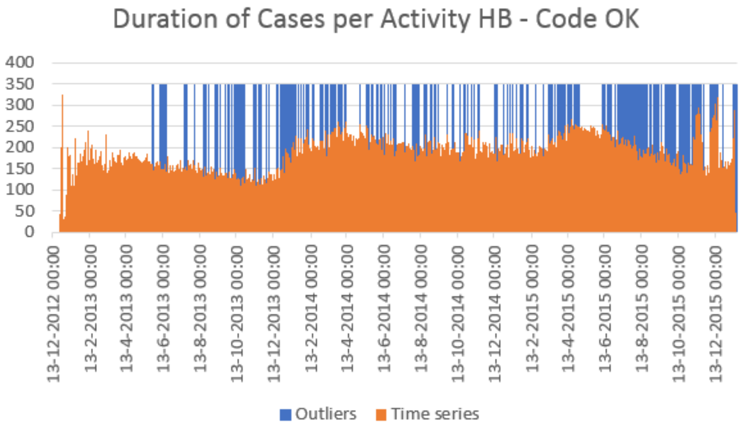


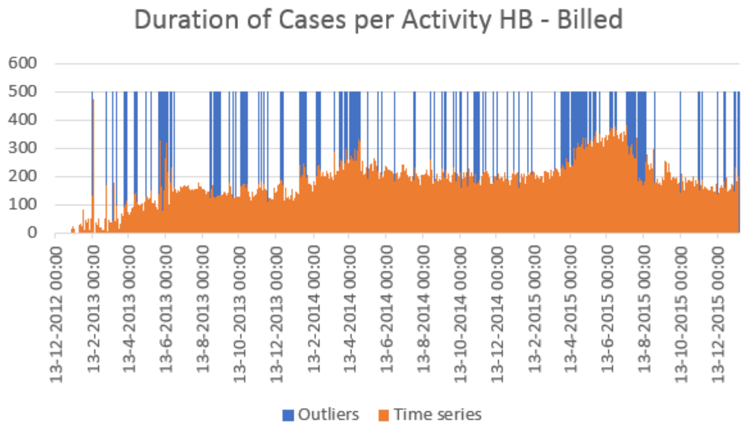
Duration of Cases per Activity HB - New

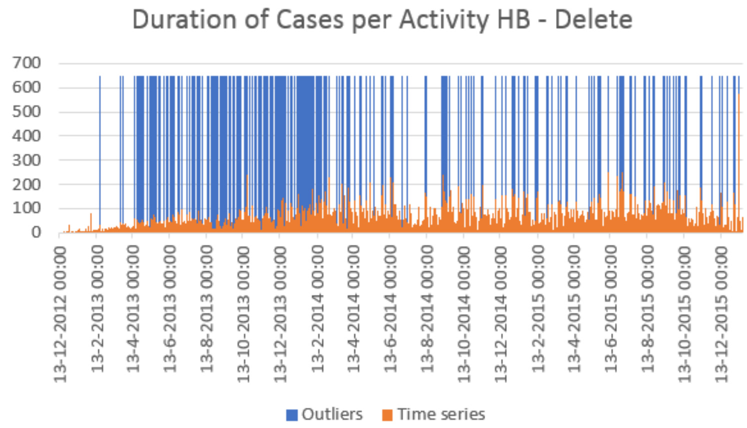


Duration of Cases per Activity HB - Fin

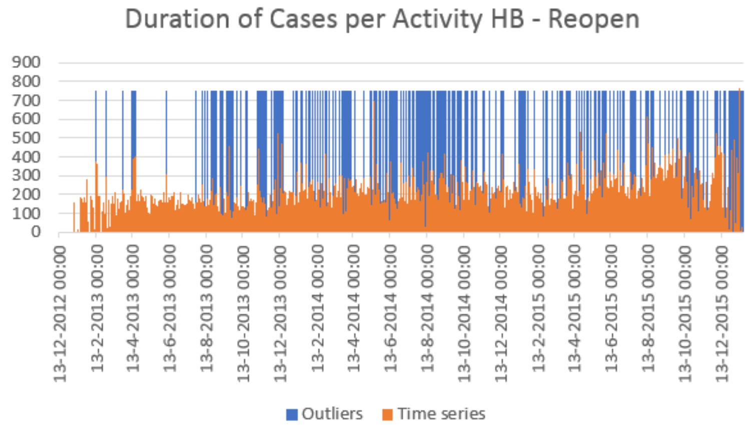

Duration of Cases per Activity HB - Release
400
350
300
250
200
150
100
50
0
13-12-2012 00:00
13-2-2013 00:00
13-4-2013 00:00
13-6-2013 00:00
13-8-2013 00:00
13-10-2013 00:00
13-12-2013 00:00
13-2-2014 00:00
13-4-2014 00:00
13-6-2014 00:00
13-8-2014 00:00
13-10-2014 00:00
13-12-2014 00:00
13-2-2015 00:00
13-4-2015 00:00
13-6-2015 00:00
13-8-2015 00:00
13-10-2015 00:00
13-12-2015 00:00
Outliers
Time series


Duration of Cases per Activity HB - Code OK
400
350
300
250
200
150
100
50
0
13-12-2012 00:00
13-2-2013 00:00
13-4-2013 00:00
13-6-2013 00:00
13-8-2013 00:00
13-10-2013 00:00
13-12-2013 00:00
13-2-2014 00:00
13-4-2014 00:00
13-6-2014 00:00
13-8-2014 00:00
13-10-2014 00:00
13-12-2014 00:00
13-2-2015 00:00
13-4-2015 00:00
13-6-2015 00:00
13-8-2015 00:00
13-10-2015 00:00
13-12-2015 00:00
Outliers
Time series


Duration of Cases per Activity HB - Billed
600
500
400
300
200
100
0
13-12-2012 00:00
13-2-2013 00:00
13-4-2013 00:00
13-6-2013 00:00
13-8-2013 00:00
13-10-2013 00:00
13-12-2013 00:00
13-2-2014 00:00
13-4-2014 00:00
13-6-2014 00:00
13-8-2014 00:00
13-10-2014 00:00
13-12-2014 00:00
13-2-2015 00:00
13-4-2015 00:00
13-6-2015 00:00
13-8-2015 00:00
13-10-2015 00:00
13-12-2015 00:00
Outliers
Time series



Duration of Cases per Activity HB - Delete



Duration of Cases per Activity HB - Reopen

D Duration of cases

The duration of cases for the outlier cases event log, the non-outlier cases event log and the complete event log will be give in table 17 until 20 for every combination of characteristic and event log.

Table 17: Duration of Cases "Events per Day"

Event log	**Outliers**	**Non-outliers**	**Entire Log**
Sepsis	36,1 days	19,3 days	28,5 days
HL	429 days	333 days	381 days
BPIC 2012	18,8 days	8,2 days	8,6 days
BPIC 2107	149 days	27,3 days	27,7 days
RTF	690 days	148 days	342 days
HB	48,2 days	34,9 days	37,8 days

Table 18: Duration of Cases "Number of Case Variant on a Day"

Event log	**Outliers**	**Non-outliers**	**Entire Log**
Sepsis	36,3 days	18,9 days	28,5 days
HL	426 days	340 days	381 days
BPIC 2012	17,2 days	7,1 days	8,6 days
BPIC 2107	354 days	27,1 days	27,7 days
RTF	435 days	19,2 days	342 days
HB	44,7 days	25,7 days	37,8 days

Table 19: Duration of Cases "Duration of Cases"

Event log	**Outliers**	**Non-outliers**	**Entire Log**
Sepsis	77,5 days	20,4 days	28,5 days
HL	723 days	378 days	381 days
BPIC 2012	14,1 days	8,5 days	8,6 days
RTF	315 days	345 days	342 days
HB	12,4 days	40,9 days	37,8 days

Table 20: Duration of Cases "Duration of Activities"

Event log	**Outliers**	**Non-outliers**	**Entire Log**
BPIC 2107	181 days	27,4 days	27,7 days