Lars Kegel, Martin Hahmann, Wolfgang Lehner

Generating What-If Scenarios for Time Series Data

Erstveröffentlichung in / First published in:

SSDBM '17: 29th International Conference on Scientific and Statistical Database Management, Chicago 27.-29.06.2017. ACM Digital Library, Art. Nr. 3. ISBN 978-1-4503-5282-6.

DOI: https://doi.org/10.1145/3085504.3085507

Final edited form was published in "SSDBM '17: 29th International Conference on Scientific and Statistical Database Management. Chicago 2017", Art. Nr. 3,
ISBN 978-1-4503-5282-6
https://doi.org/10.1145/3085504.3085507

Generating What-If Scenarios for Time Series Data

Lars Kegel
Technische Universität Dresden
01062 Dresden, Germany
lars.kegel@tu-dresden.de

Martin Hahmann
Technische Universität Dresden
01062 Dresden, Germany
martin.hahmann@tu-dresden.de

Wolfgang Lehner
Technische Universität Dresden
01062 Dresden, Germany
wolfgang.lehner@tu-dresden.de

ABSTRACT

Time series data has become a ubiquitous and important data source in many application domains. Most companies and organizations strongly rely on this data for critical tasks like decision-making, planning, predictions, and analytics in general. While all these tasks generally focus on actual data representing organization and business processes, it is also desirable to apply them to alternative scenarios in order to prepare for developments that diverge from expectations or assess the robustness of current strategies. When it comes to the construction of such what-if scenarios, existing tools either focus on scalar data or they address highly specific scenarios. In this work, we propose a generally applicable and easy-to-use method for the generation of what-if scenarios on time series data. Our approach extracts descriptive features of a data set and allows the construction of an alternate version by means of filtering and modification of these features.

CCS CONCEPTS

• **Mathematics of computing** → **Time series analysis**; • **Computing methodologies** → **Model development and analysis**; • **Information systems** → *Data mining*;

KEYWORDS

what-if scenario, what-if analysis, hypothetical query, time series analysis, business analytics

ACM Reference format:
Lars Kegel, Martin Hahmann, and Wolfgang Lehner. 2017. Generating What-If Scenarios for Time Series Data. In *Proceedings of SSDBM'17, Chicago, IL, USA, June 27-29, 2017,* 12 pages.
https://doi.org/10.1145/3085504.3085507

1 INTRODUCTION

What-if scenarios are simulations whose goal is to check a system under some given hypotheses. They are applied in a multitude of application domains and support decision-making and planning. Typically, their generation comes down to varying parameters in a target function or systematically modifying values of a data set, applied to data types such as spreadsheets or OLAP cubes [18].

DOI: https://doi.org/10.1145/3085504.3085507

In our research activities, we focus on capturing time series features for evaluation and application purposes. Time series describe the dynamic behaviour of a monitored object or process over time. They arise, for example, from consumption in the energy domain, from sales figures in market research, and from sensor readings in manufacturing processes.

In the energy domain, analysts aim for grid balancing and avoiding costly grid upgrades. They generate what-if scenarios to assess the risk of over- or under-consumption by the energy grid due to, for instance, a changed user behavior, population growth, the establishment of a new manufacturing site, or consequences of a black-out. Practically, they carry out a measurement analysis: taking past measurements and a set of hypotheses, they can study future measurements and their consequences. Thus, they could compare a hypothesis a) "Future measurements will follow the forecast" against a hypothesis b) "We assume that future measurements are 10% higher than forecasts produced from 5% lower measurements" as presented in [26]. The hypothesis b) includes a what-if scenario.

In market research, a common task is the discussion of sales figures in order to assess the success of a company. What-if scenarios support assessments like, e.g., "How would sales figures look like under different market assumptions?" and different pricing scenarios for different products can be discussed [5].

In manufacturing, users want to figure out the key performance indicators of a manufacturing site. This relates to decisions in capacity planning and maintenance. For planning purposes, a valid question would be, "Can the production facility keep up with a different market situation?" Whereas for maintenance, users would like to know, for example, "What if a production line has to produce a higher number of lots? When will a machine wear out and when is a failure likely to occur?" Clearly, what-if scenarios on time series can be beneficial in many different domains.

Still, little attention has been paid to conceptualize the generation of what-if scenarios for time series. Either they are too general (i.e., they do not take time series features into account [26]) or they are too specific (i.e., they involve highly sophisticated models for a limited use case [7, 8]). Our goal is a time series representation for creating what-if scenarios that are applicable in a multitude of application domains.

The method that we present consists of an analytical and an interactive part. It automatically decomposes time series into components which are subsequently characterized by scalar values, called features. We will show that these features are a promising representation for time series components. The second part describes the interaction which enables users to visually modify time series data.

Section 2 surveys related work from time series analysis and what-if scenarios. Section 3 presents our approach as a workflow

Final edited form was published in "SSDBM '17: 29th International Conference on Scientific and Statistical Database Management. Chicago 2017", Art. Nr. 3,
ISBN 978-1-4503-5282-6
https://doi.org/10.1145/3085504.3085507

of analytical and interactive steps. In Section 4, we present three what-if scenarios in a real-world use case, followed by a discussion in Section 5. Finally, we conclude our work in Section 6.

2 RELATED WORK

In order to apply what-if scenarios on time series data, we need to extract time series features which represent their shape and behavior. We discuss these transformations first in this section. Subsequently, we review what-if scenarios and their role as a pre-requisite for business analytics. Finally, we present two example data sets that we use for our explanations. To the best of our knowledge, what-if scenarios on time series features have not been adressed by the database or statistics community.

2.1 Decomposition of Time Series

Time series form an ubiquitous data type and occur in a multitude of domains. Capturing the correlation of adjacent values has most often been studied and is commonly referred to as time series analysis.

Most existing techniques describe a time series as a combination of three components: a *trend*, a *season* and a *residual* component [23]. The trend represents the long-term change in the mean level of the series, whereas the season describes a cyclical repeated behavior. Residuals usually represent unstructured information that is generally assumed to be random. The sum of these three components is called *additive model* and represents economic and energy time series [23].

Knowing the components, we now look at decomposition techniques that can extract them. There are several techniques based on moving-average models or regression: *classical decomposition* and the more sophisticated methods, *X-13* and *STL*, that we subsequently discuss.

Classical decomposition dates back to the 1920s and is the basis for most subsequent decomposition techniques [9, 12]. The key concept is to retrieve the trend by applying a moving-average process on given time series. Afterwards, the season is calculated by averaging the detrended measures of associated time instances: in case of monthly values, all values of January are averaged, all measures of February, and so on. The season may be of arbitrary *season length L*. It is stable, i.e., the seasonal pattern does not change from season to season. In case of monthly values, the seasonal value for January is constant throughout all years, the seasonal value for February, and so on. A major drawback is that this method does not decompose the first and last values of the time series, called *endpoints*, due to the moving average filter. Consequently, components cannot be completely retrieved.

In the 1970s, the X-11 method from the U.S. Bureau of the Census was published and adopted by several statistical agencies around the world [4]. This method and its successors, X-12 and X-13, furthered the concept of classical decomposition with several moving-average steps [1, 9]. Most importantly, they use predictions from forecast models backwards and forwards in time such that the endpoints can be decomposed, too. Slowly varying seasons are possible, they represent changes in the seasonal behavior. Nevertheless, the methods are designed for decomposing only quarterly and monthly data that is why this method is not applicable in our general approach.

Table 1: Features of Decomposition Techniques

	DEC	X-13	STL
Arbitrary season length	✓	-	✓
Slowly varying season	-	✓	✓
Robustness	-	✓	✓
Decomposition of endpoints	-	✓	✓

In the 1990s, Cleveland [3] found that *Loess smoothing*, a locally-weighted regression technique, also leads to good results for detrending and deseasonalizing a time series. His method, STL, is considered as a versatile and robust decomposition technique, handling every type of season length and decomposing endpoints [9]. Since this method is widely and recently applied [25], we adopt it in our approach.

Table 1 summarizes the features of the presented techniques. STL fulfills all criteria that we require for our automatical and generally applicable transformation. Thus, we adopt this this technique for our prototype.

2.2 Features of Time Series

Time series and their components have a high dimensionality due to their length. We aim for reducing them to scalar values, so called *features*, that represent their characteristics.

Common features are *minimum*, *maximum* and the central moments *mean* and *standard deviation*. Intuitively, a what-if scenario modifies a time series such that its extreme values and moments are shifted. As an example, the linear scale $(0, 1)$ transforms the values to the given interval, i.e., the minimum value is 0 and maximum is 1. The z-score transforms the values such that the mean value is 0 and the standard deviation is 1. These are features that are applicable on time series in general.

More relevant to our work are features related to time series components. Wang et al. [27] present features in their work on time series clustering techniques. They define a *trend* and *season determination* based on the coefficient of determination. These features describe the influence of the respective components compared to the residuals on a scale from 0 to 1, where 0 means no influence of the component and 1 means highest. It seems very intuitive and easy to understand that features represent the influence of a time series component on its time series. Therefore, we apply and extend this concept in our work which we will show in more detail in Section 3.

2.3 What-If Questions in Analytics

Traditionally, what-if scenarios have been presented in the context of spreadsheets and OLAP tools. Rizzi [18] defines them as "data-intense simulations whose goal is to inspect the behavior of a complex system [...] under some given hypotheses." What-if scenarios for spreadsheets allow for higher analytical tasks and are integrated in an interactive environment. However, they lack storage capacity and performance. OLAP tools are complementary in that they offer a better storage and performance but only support basic analytical tasks.

Final edited form was published in "SSDBM '17: 29th International Conference on Scientific and Statistical Database Management. Chicago 2017", Art. Nr. 3,
ISBN 978-1-4503-5282-6
https://doi.org/10.1145/3085504.3085507

Since then, several tools adopt this concept, such as SAP Strategy Enterprise Management, SAS Forecast Server and Microsoft Analysis Services [18]. Oracle offers basic support for what-if scenarios by implementing the MODEL clause in its data warehouse [15, 20]. These tools deal with modification and analytics on OLAP data.

More recently, Evans identifies what-if questions as a major requirement for business analytics [5]. He defines business analytics as "the use of data, information technology, statistical analysis [...] to help managers [...] make better, fact-based decisions". Moreover, he identified two main reasons why the analytics market is increasing in terms of bigger analytics departments, richer university programs and wider research literature: A) Studies have shown that companies are performing better in terms of profitability when they invest in analytics departments that support their fact-based decisions. B) Companies are more and more overwhelmed with larger data set they retrieve from their processes and they require a better understanding of how to turn this data into insights.

Evans views business analytics from three different perspectives: a descriptive, a predictive, and a prescriptive perspective. Descriptive analytics is the first and most common perspective, it analyzes past and current data in order to prepare informed decisions. Its techniques are among others the consolidation, the classification, and the reporting of data. Questions that may be answered by these techniques are concentrated on the past like, for example, "What was the revenue of product x in year y?" or "Which factory has the lowest performance?"

Predictive analytics goes a step further in that it takes future data into account and assesses future behavior. Thus, its techniques are the prediction and extrapolation of data. Questions that arise are typically focused on these predictions like, for example, "What will happen if the market figures continue increasing as they did last year?" Thus, predictive analytics is built on top of descriptive analytics: it uses past data and links it with forecasting models for future assessments.

Prescriptive analytics is the third phase of business analytics which goes beyond descriptive and predictive analytics. Not only takes this perspective past data and predictions into account, it also uses optimization techniques in order to suggest actions and decisions with respect to a given goal function. By this means, it supports users to discover and to take better, fact-based decisions and to answer question like, for example, "How much should we produce a product x for maximizing the profit?"

There are tools that support these analytics tasks and that arise in different domains such as statistics, modeling, optimization, and business intelligence. What-if questions are among these tools in that they are an intersection of business intelligence and modeling. They link a number of input variables, which are assumptions given by users, with an output value, which is a what-if scenario. Based on past and current data, what-if scenarios are one requirement in order to assess predictive and prescriptive analytics. They enable users to build forecasting models based on hypothetical assumptions and to assess the effects and actions to take if these assumptions become true.

Since time series are an important data type, our work focuses on a description of what-if scenarios for time series. By this means, we contribute an approach that may be further used by predictive and prescriptive analysis tools for decision-making.

Table 2: Time Series Components of M3-Competition

	No Season	Season	Sum
No Trend	209	58	267
Trend	2122	614	2736
Sum	2331	672	3003

Table 3: Time Series Components of Smart Metering Project

	No Weekly	Weekly	Sum
No Yearly	-	-	-
Yearly	1934	2687	4621
Sum	1934	2687	4621

2.4 Example Data Sets

Throughout this paper, we explain our approach using two running examples.

Example 2.1 (M3-Competition). The M3-Competition is the latest of three M-Competitions in 2000 [14]. Its goal is the systematic evaluation of forecast method accuracy on a defined data set. The data set consists of 3003 time series that are from different origins (industry, finance, demographic, macro-/microeconomic, other). The values of each time series have a defined interval (year, quarter, month, other). Time series of the M3-Competition exhibit a trend, a seasonal component or both of them (Table 2).

Example 2.2 (Smart Metering Project). The Irish Commission for Energy Regulation initiated the Smart Metering Project in order to assess the performance of smart meters in Irish households and businesses [24]. They measured the consumption between July 2009 and December 2010 and made the data set available in an anonymized format, indicating code (specifying residential, small or medium business (SME), other), smart meter ID, timestamp and consumption. We choose 4621 time series that are complete and whose code was publicly available and aggregated them to daily values. These time series have no trend but a weekly and a yearly season. Since the time interval of the data set is shorter than two years, the yearly season cannot be extracted as a season component. Instead, it is considered as a long-term change. Hence, the data set has the components as given in Table 3.

3 OVERVIEW OF OUR APPROACH

In this section, we explain the methods that we apply in our approach. To highlight the major steps, Figure 1 shows an overview as a flowchart. We begin our description with the analytical steps and conclude with the visualization and interaction step. A *time series relation* stores time series in a *database*. Its structure is described in more detail in Subsection 3.1. After retrieving the data, time series are transformed. The goal of the *transformation* is to derive *time series components* and to reduce components to common

Final edited form was published in "SSDBM '17: 29th International Conference on Scientific and Statistical Database Management. Chicago 2017", Art. Nr. 3,
ISBN 978-1-4503-5282-6
https://doi.org/10.1145/3085504.3085507

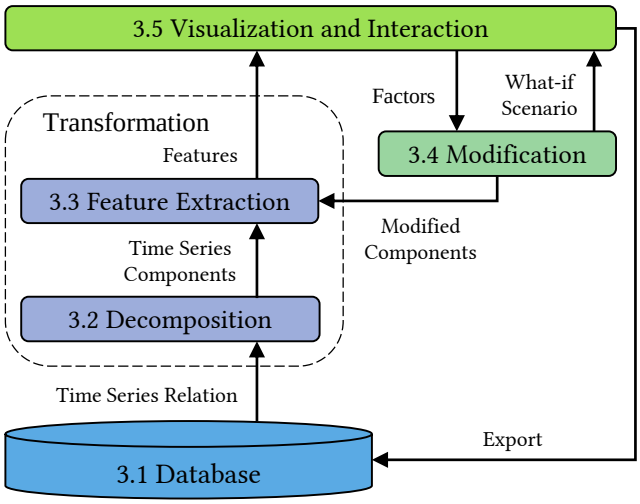


Figure 1: System Overview

features. Thus, this step addresses two tasks: *decomposition* and *feature extraction*, which are explained in Subsections 3.2 and 3.3. A feature space further visualizes features and allows for interaction by the user. By giving *factors*, time series components and features are subsequently modified. This *modification* step is presented in Subsection 3.4 and results in a *what-if scenario*. Moreover, *modified components* update the respective features. The components of *visualization and interaction* are explained in Subsection 3.5. We *export* scenarios back to the database. Finally, we give an overview of the implementation in Subsection 3.6.

3.1 Database

Our goal is to build the approach on top of a database with a unified time series representation. This representation includes time and measurement information as well as categorical information. We tackle this by adopting the time series relation from Fischer [6].

Naturally, time series are defined by its strict order over a *time dimension*. A time dimension is characterized by a set of *time attributes*, whose composition forms a *time domain*. A value in the time domain is a *time instance*, a set of consecutive values forms a *time interval*. For example, a tuple (1, 2016) is a time instance from the time domain (Month, Year), meaning January 2016. Thus, a time series in the relation model is defined as follows:

Definition 3.1 (Time Series). A time series over a time interval is formed by a set of tuples over a schema consisting of

- One or more time attributes with instances from the time domain,
- One or more *measure attributes* with real-valued measurements, so-called *values*.

In our method, time series are equidistant and complete, which means that there are no null values. Time instances are unique. Subsequently, Fischer defines a time series relation as follows:

Definition 3.2 (Time Series Relation). A set of time series composes a time series relation over a schema consisting of *category attributes*, time and measure attributes.

Table 4: Example Time Series Relation

Code	Meterid	Date	Consumption
SME	1050	2009-07-14	36.809
SME	1050	2009-07-15	34.941
SME	1050	2009-07-16	32.477
Residential	1052	2009-07-14	15.206
Residential	1052	2009-07-15	11.256
Residential	1052	2009-07-16	19.829
…	…	…	…

- A time series relation has zero or more category attributes that are of arbitrary domain and that are invariant with respect to time.
- Each time series has equal values in all category columns. Each distinct set of category attributes determines exactly one time series.
- Each time series follows the properties of Def. 3.1.

Table 4 represents a time series relation for our running example (Smart Metering Project). The *meterid* uniquely describes a given time series. *Code* is a category attribute, *date* is the time attribute and *consumption* the measure attribute.

3.2 Decomposition

We suppose that time series are a combination of a trend, a season, and residual component. Whether it contains a trend or season component has to be checked first. In our automatic environment, this is carried out with test methods. The possible combinations and the applied techniques are shown in Table 5.

The trend check is done by extracting the long-term mean of the time series and by testing whether this mean is a trend. We use a kernel smoothing method to extract the long-term mean. Presented by [16, 19], this method is based on moving-average smoothing with a weight function to average observations. A bandwidth parameter b configures how smooth the result is. The method is selected because it smooths also the endpoints of a time series which is not the case for usual moving-average filters. We refer to the bandwidth parameters $b = 2$ given by [23] that smooths a long-term mean.

Subsequently, the trend test of Mann-Kendall [13] enables us to check whether the long-term mean is considered as a trend. The null hypothesis states that there is no trend in the sample, whereas the alternative hypothesis states that there is a trend. We accept the null hypothesis according to a significance level $\alpha = 0.05$. Otherwise, we accept the alternative hypothesis.

Second, we check for a seasonal behavior on the detrended series. For this purpose, we rely on the procedure presented by Wang et al. [27] with one modification. The autocorrelation function of the detrended series returns autocorrelation coefficients for all lags up to 1/3 of the series length. Peaks and lows are visible and show which lag has the highest autocorrelation. The season length is the lag of the first peak of a positive autocorrelation that is preceded by a low. As we only want to assert the existence of a season, we

Final edited form was published in "SSDBM '17: 29th International Conference on Scientific and Statistical Database Management. Chicago 2017", Art. Nr. 3,
ISBN 978-1-4503-5282-6
https://doi.org/10.1145/3085504.3085507

Table 5: Decomposition Techniques

	Season	No Season
Trend	STL	Kernel Smoothing with $b = 2$
No Trend	Kernel Smoothing with $b = 5/L$	-

modify this method in that we only accept the lag as season length if

- the autocorrelation difference between peak and low is at least 0.1 [27],
- the autocorrelation is significant in that it is within the confidence interval and positive,
- the lag is greater than 1 and it is a multiple of the frequency given by the data or vice-versa (monthly values only accept a lag 6, 12, 24 etc.).

The lag that is returned either confirms the season length given by the data or it is 1 (no season). If only a season but not a trend component exists we smooth the season with bandwidth $b = 5/L$ (where L is the season length) as given by [23]. If no season component exists but a trend, the trend is the extracted long-term mean.

If a trend as well as a season component exist, we choose STL as decomposition method because it is robust and applicable on arbitrary season lengths. Full details of the method are given in [3]. All parameters were selected automatically except for the smoothness of seasonal subcycles. That smoothness represents the variation from one season to another. The greater the value the smoother are the subcycles which means there is less variation. As it highly depends on the knowledge of the time series, users must carry out some diagnostics first. To avoid this task in an automatic context, we assume that seasons do not vary from one cycle to another and are highly robust.

The process in Figure 2 is the time series named N1906 from the M3-Competition along with its components trend, season, and residuals. It measures the number of recreation visits in national parks from January 1983 to August 1992, thus, it belongs to the category "Industry". Its trend component shows an increase of 800 visits on average along these 10 years. Fluctuations due to less and higher visits are also captured which results in a trend that deviates from a strongly linear behavior. The season component has a high influence because its range (from -3100 to 5500) is 10 times higher than the trend increase. Since the decomposition ensures a high robustness, the season does not vary from one season to another.

3.3 Feature Extraction

Time series are further transformed by extracting features from the components. Three trend features are chosen: determination, slope and linearity. Additionally, we choose the season determination. We explain these features with respect to the additive model. Let x_t be the original time series, then

$$x_t = tr_t + seas_t + res_t \tag{1}$$

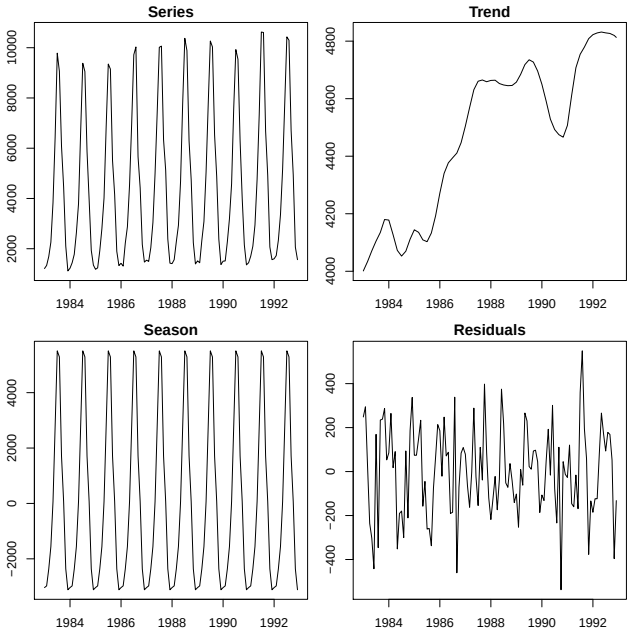


Figure 2: Decomposition with STL: Series and Trend, Season and Residuals

where tr_t is the trend, $seas_t$ the season, and res_t the residual component.

3.3.1 Trend Determination. According to [27], the *trend determination* represents the influence of the trend component on the time series. The coefficient of trend determination is then

$$R^2_{tr} = 1 - \frac{var(res_t)}{var(res_t + tr_t)} \tag{2}$$

where $var(y_t) = \frac{1}{T-1}\sum_{t=1}^{T}(y_t - \bar{y})^2$ is the sample variance, y_t $(1 \leq t \leq T)$ is a sample of T values and $\bar{y}$ is the sample mean. The trend determination ranges between 0 and 1: $R^2_{tr} = 0$ means that the series is determined by the residuals and the influence of the trend is negligible, whereas $R^2_{tr} = 1$ shows a high trend influence. Thus, this feature is promising because users can examine the data set under the assumption of a weaker or stronger trend. Time series N1906 has a moderate trend determination, $R^2_{tr} = 0.66$.

3.3.2 Trend Slope. We assume that there is a linear trend. Thus, there is a *trend slope* that captures an overall increase or decrease of the time series, whereas the trend component tr_t derived by the STL captures also local trend changes. In an attempt to identify the slope, we fit a linear regression model to tr_t:

$$tr_t = \theta_1 + \theta_2 \cdot l_t + \delta_t \tag{3}$$

such that the sum of the squares of δ_t is minimal.

The slope is represented by θ_2, it can be retrieved and manipulated for generating what-if scenarios. A high slope results in a high increase of the trend whereas a slope near 0 means that there is no overall increase. Given the time series N1906, the slope is $\theta_2 = 7.24$ which means an increase by 7.24 visits per month on average.

Final edited form was published in "SSDBM '17: 29th International Conference on Scientific and Statistical Database Management. Chicago 2017", Art. Nr. 3,
ISBN 978-1-4503-5282-6
https://doi.org/10.1145/3085504.3085507

The other parameters of linear regression are as follows. θ_1 is the base value from which the trend starts. The base value of time series N1906 is $\theta_1 = 4044.74$.

l_t is the vector of time instances. For our calculations, we suppose that l_t is a sequence of integers $[0, 1, \dots]$. For visualization purposes, this is mapped to the time instances as given by the time series. In case of N1906, this means [Jan 1983, Feb 1983, . . . , Aug 1992].

The difference between a trend from STL and from linear regression is expressed as δ_t, representing the difference of local trend changes in STL compared to the overall trend behavior.

3.3.3 Trend Linearity. The *trend linearity* expresses the relation between the linear regression model (3) and the trend component. Indeed, a trend that has little variation, has a strong linearity. This feature is captured by

$$R^2_{lin} = 1 - \frac{var(\delta_t)}{var(tr_t)} \tag{4}$$

$R^2_{lin} = 1$ means that the trend is a straight line and the residuals δ_t are negligible. Otherwise, the trend fluctuates. In case of time series N1906, the feature is $R^2_{lin} = 0.80$ which indicates, that there are some slight variations as shown in Figure 2.

3.3.4 Season Determination. The *season determination* represents the strength of the season component on the time series [27]. Analogous to the trend, the coefficient of season determination is

$$R^2_{seas} = 1 - \frac{var(res_t)}{var(res_t + seas_t)}. \tag{5}$$

This allows to generate what-if scenarios where the seasonal fluctuation is increased or diminished. Given the time series N1906, the season determination is almost 1.00 and confirms the strong influence of the respective component.

Concluding the transformation, the data set consists of time series components tagged with its respective features. On the one hand, this enables users to filter not only by category attributes but also by features. On the other hand, it enables them to modify features, as subsequently explained.

3.4 Modification

A what-if scenario consists of data with hypothetical modifications. In our scenario, features are representatives of time series components and we propose their modification by introducing factors. Factors may increase or decrease the feature proportionally or non-linearly. We reuse the example time series N1906 from M3-Competition (Figure 2) and give scenarios and the dependency from factors for the aforementioned features (Figure 3). Additionally, we allow for adding linear trends.

3.4.1 Trend Determination Factor. Let f be a factor that varies trend determination. The modified trend $tr_{t,f}$ is defined by:

$$tr_{t,f} = \theta_1 + f \cdot (\theta_2 \cdot l_t + \delta_t) \tag{6}$$

This equation represents the linear regression model that is fitted to the trend. f is a factor applied to the slope θ_t and the difference δ_t. Depending on f, the trend determination increases ($f > 1$), decreases ($0 \leq f < 1$), or is left unchanged ($f = 1$). A factor $f < 0$ is not admissible.

The effect of this factor is represented by Figure 3(a). The plot shows the original trend (blue with triangles) and three modified trends. The latter ones are modified by a trend determination factor $f = 1.25$, $f = 0.75$, and $f = 0.50$, respectively. Overall, the main characteristics of the trend are kept but they are a multiple of the former value. In January 1988, the given trend value is 4663, the trend value with factor $f = 0.5$ is 4354. The influence on the trend determination R^2_{tr} is given in the figure's legend. It is shown that the trend determination factor and the trend determination is not proportional.

This becomes clear with Figure 3(b) on the given time series. It presents the trend determination R^2_{tr} for factors f between 0.00 and 2.00. By increasing f, determination approaches 1. The slope of this figure depends on the variance of the time series components.

3.4.2 Trend Slope Factor. Let g be a factor that varies the trend slope. The modified trend $tr_{t,g}$ is defined by:

$$tr_{t,g} = \theta_1 + g \cdot \theta_2 \cdot l_t + \delta_t \tag{7}$$

Again, we apply the factor to the linear regression model. But in this case, only the slope is modified and not the difference δ_t. Depending on g, the trend slope increases ($g > 1$), decreases ($0 \leq g < 1$), or is left unchanged ($g = 0$). A factor $g < 0$ is not admissible.

This effect is represented in Figure 3(c). Again, the original trend (blue with triangles) and three modified trends (with factors $g = 2.00$, $g = 0.75$, $g = 0.50$) are shown. All time series start at the same base level and they keep the same trend changes but their directions are different. The factor g and the trend slope are proportional.

3.4.3 Trend Linearity Factor. Local trend changes are captured by the decomposition method STL. We define the trend linearity as the determination of trend changes and of the linear trend. Let h be a factor that varies the trend linearity. The modified trend $tr_{t,h}$ is defined by:

$$tr_{t,h} = \theta_1 + \theta_2 \cdot l_t + \frac{1}{h} \cdot \delta_t \tag{8}$$

Depending on h, the linearity increases ($h > 1$), decreases ($0 \leq h < 1$) or is left unchanged ($h = 1$). A factor $h < 0$ is not admissible.

The effect of this factor represented in Figure 3(d). Again, the original trend (blue with triangles) and three modified trends (with factors $h = 1.50$, $h = 1.25$, $h = 0.75$) are shown. If the factor h increases the resulting trend is more linear because the difference δ_t is diminished.

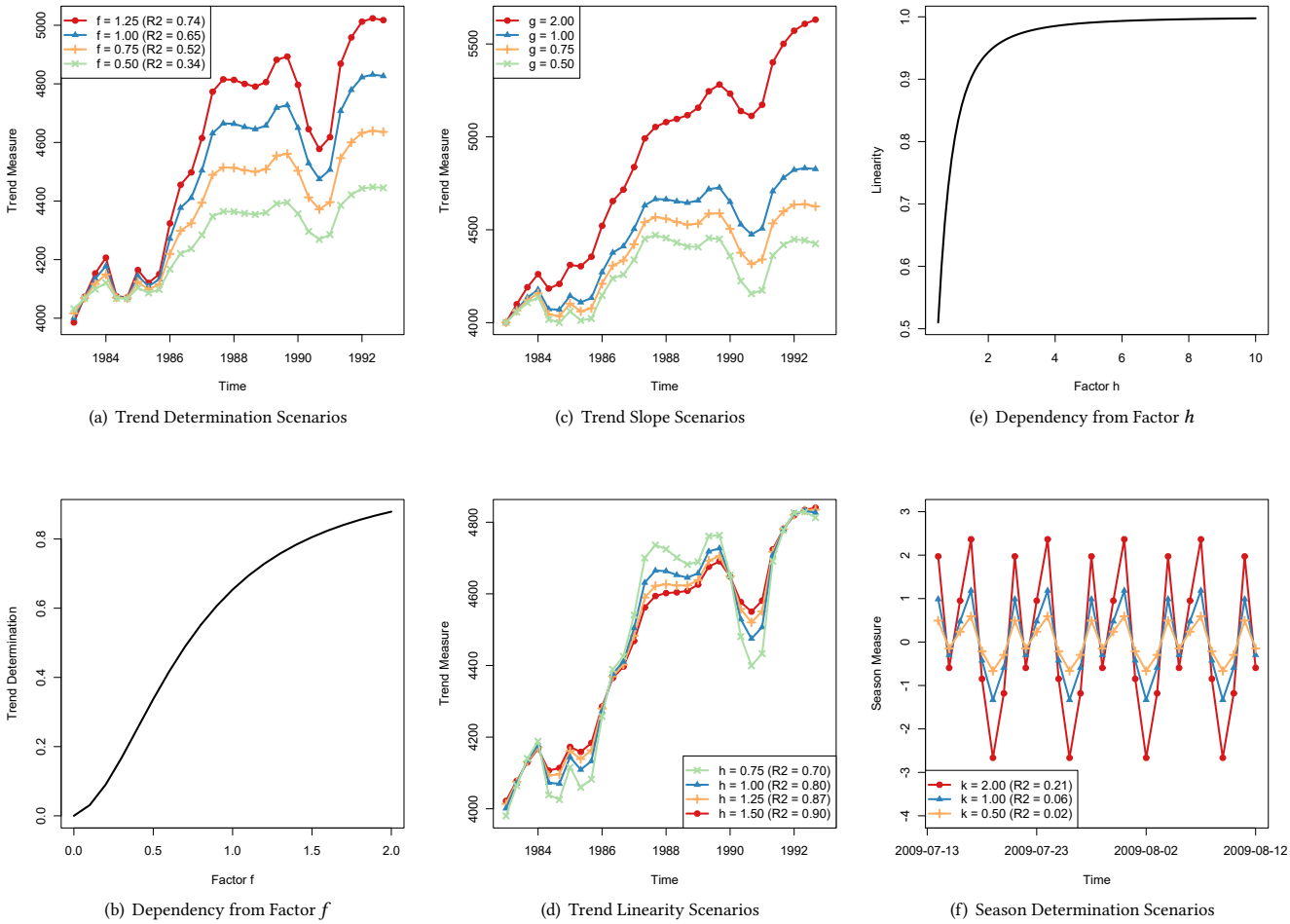
The linearity is calculated approximately with

$$R^2_{lin} = 1 - \frac{\frac{1}{h^2} \cdot var(\delta_t)}{\theta_2^2 \cdot var(l_t) + \frac{1}{h^2} \cdot var(\delta_t)} \tag{9}$$

supposing that the sample covariance of l_t and δ_t is negligible because these processes are independent. It follows that, if $h \to 0$, the trend is very noisy and not determined. If $h \to \infty$, the trend tends to be linear which means a determination of 1. Thus, the linearity factor and the trend linearity are also not proportional. This is also confirmed by Figure 3(e) which shows the trend linearity for different $0 < h < 10$.

3.4.4 Additional Trend. The aforementioned factors are applied on trends, but they do not carry out a modification on time series without this component. Introducing such a trend for assessing

Final edited form was published in "SSDBM '17: 29th International Conference on Scientific and Statistical Database Management. Chicago 2017", Art. Nr. 3,
ISBN 978-1-4503-5282-6
https://doi.org/10.1145/3085504.3085507



(a) Trend Determination Scenarios

(c) Trend Slope Scenarios

(e) Dependency from Factor h

(b) Dependency from Factor f

(d) Trend Linearity Scenarios

(f) Season Determination Scenarios

Figure 3: Factors for Modification

possible consequences is crucial for risk-analysis. Thus, we consider an *additional trend* $tr_{t,m}$ and extend our tool set by an offset m such that

$$tr_{t,m} = \theta_1 + m \cdot \theta_1 \cdot l_t \tag{10}$$

where θ_1 is the base level and l_t are the time instances. These values are calculated from the residual component in the absence of a trend component.

Subsequently, we add an offset $m \cdot \theta_1$ for each time instance. For a time series with monthly values, $m = 0.10/12$ means that the trend increases by 0.83% per month (10% per year). An application scenario in the following section uses an additional trend (Figure 6).

3.4.5 Season Determination Factor. Let there be a factor k that varies the season determination. The modified season $seas_{t,k}$ is defined by:

$$seas_{t,k} = k \cdot seas_t \tag{11}$$

Depending on k, the season determination increases ($k > 1$), decreases ($0 \leq k < 1$), or is left unchanged ($k = 1$). A factor $k < 0$ is not admissible.

This effect is represented by Figure 3(f). The plot shows the original season (blue with triangles) and two modified season components from the Smart Metering Project. The latter ones are modified by a season determination factor $k = 2.00$ and $k = 0.50$, respectively. Modifying the season by factor k leads to higher peaks and lows. The resulting R^2_{seas} is given in the legend. Again, the season determination is not proportional to k because with increasing k, R^2_{seas} approaches 1.

3.5 Visualization and Interaction

To provide an easy way of creating what-if scenarios, we propose a visual exploration which permits users to select time series by features and categorical information, to modify components by setting factors and to display the resulting what-if scenarios.

Final edited form was published in "SSDBM '17: 29th International Conference on Scientific and Statistical Database Management. Chicago 2017", Art. Nr. 3,
ISBN 978-1-4503-5282-6
https://doi.org/10.1145/3085504.3085507

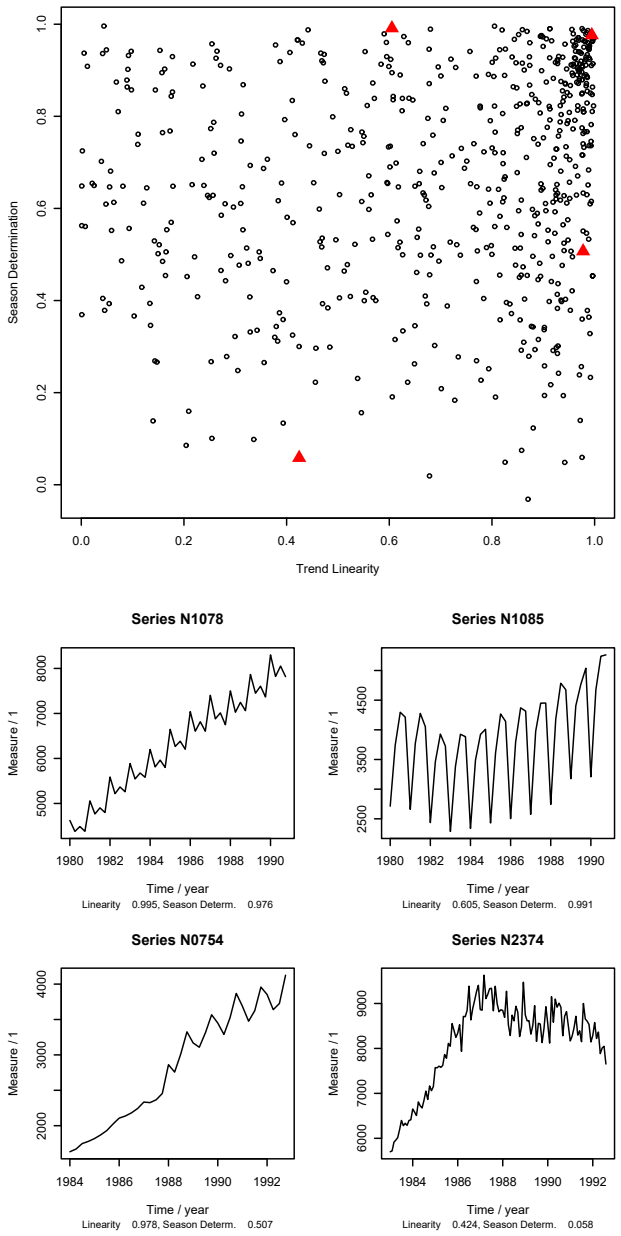


Figure 4: Trend Linearity and Season Determination: Scatterplot and Selected Time Series

Features form dimensions by which time series are sorted. Together, these dimensions build a *feature space* which is described by Kang et al. [10]. We focus on visualizing one or two features at a time. A one-dimensional feature space is visualized with a histogram, whereas a two-dimensional feature space is visualized with a scatter plot. We further the idea of Kang et al. in that users interact with the instance space. By clicking and brushing, users select time series that are subject to further modification.

Figure 4 shows the instances of the M3-Competition in a two-dimensional scatterplot. The axes show the trend linearity and the season determination. Every dot represents a time series. We choose four time series (red triangles) which exhibit different features and which are also plotted as a common line plot. Series N1078 clearly shows a strong season (compared to its residuals) and a very linear behavior. Therefore, it is in the upper right corner of the scatterplot. Series N1085 is less linear due to a trend which is not constantly increasing. While series N0754 is still very linear, it does not exhibit a strong season. Finally, the series N2374 does not exhibit any of these two features. Thus, users get an insight of a) how the time series are spread across the instance space and b) which are the time series that reside in a certain feature range.

Users generate a what-if scenario by setting factors for one or two features. Affected time series are then recalculated and their instance point is moved to the resulting position. Modifications are carried out consecutively which means that the ordering of modifications is important. The visualization is limited in that there are only two features represented together and thus, modifications are limited. Subsequent modifications are possible by changing the displayed feature after a modification step and re-applying a modification. In conclusion, we provide a visualization that allows for a user-friendly interaction and that covers the presented steps for generating what-if scenarios.

3.6 Implementation

The generation of what-if scenarios is implemented as the R package *whatif*, reusing several statistical methods from R [17]. It is available online[1].

Figure 5 gives an overview of the visualization using the M3-Competition. Four boxes are displayed: the *feature space* shows a scatter plot of the features, a *time series summary* shows a line plot with original time series and the modified time series (if available). Below, users may set features that are displayed (*select axis*) and select *modifications* that lead to the desired what-if scenario. In order to make sure that a series does not exceed an admissible range, users may fix a value domain in the modification dialog, too, consisting of a minimum and a maximum value. In this example, time series instances whose trend determination and linearity is between 0.50 and 0.75 have been selected and shifted by a factor $f = 1.2$ and $h = 1.2$. This results in a shift of the instances to a higher determination and linearity value as shown in the feature space (red triangles).

4 WORKING WITH WHAT-IF SCENARIOS

Our goal is the generation of scenarios for evaluating time series data under different assumptions. This allows users to check decision-making and planning and to assess a system's robustness.

We discuss an application of what-if scenarios in the energy domain which is based on the Smart Metering Project. We did not choose the M3-Competition because these time series originate from different processes and their aggregation would be meaningless. Subsection 4.1 outlines the transformation and modification steps for this data set. In Subsection 4.2, we introduce and discuss three example cases which may be expressed by what-if scenarios.

[1] https://lkegel.shinyapps.io/whatif/

Final edited form was published in "SSDBM '17: 29th International Conference on Scientific and Statistical Database Management. Chicago 2017", Art. Nr. 3,
ISBN 978-1-4503-5282-6
https://doi.org/10.1145/3085504.3085507

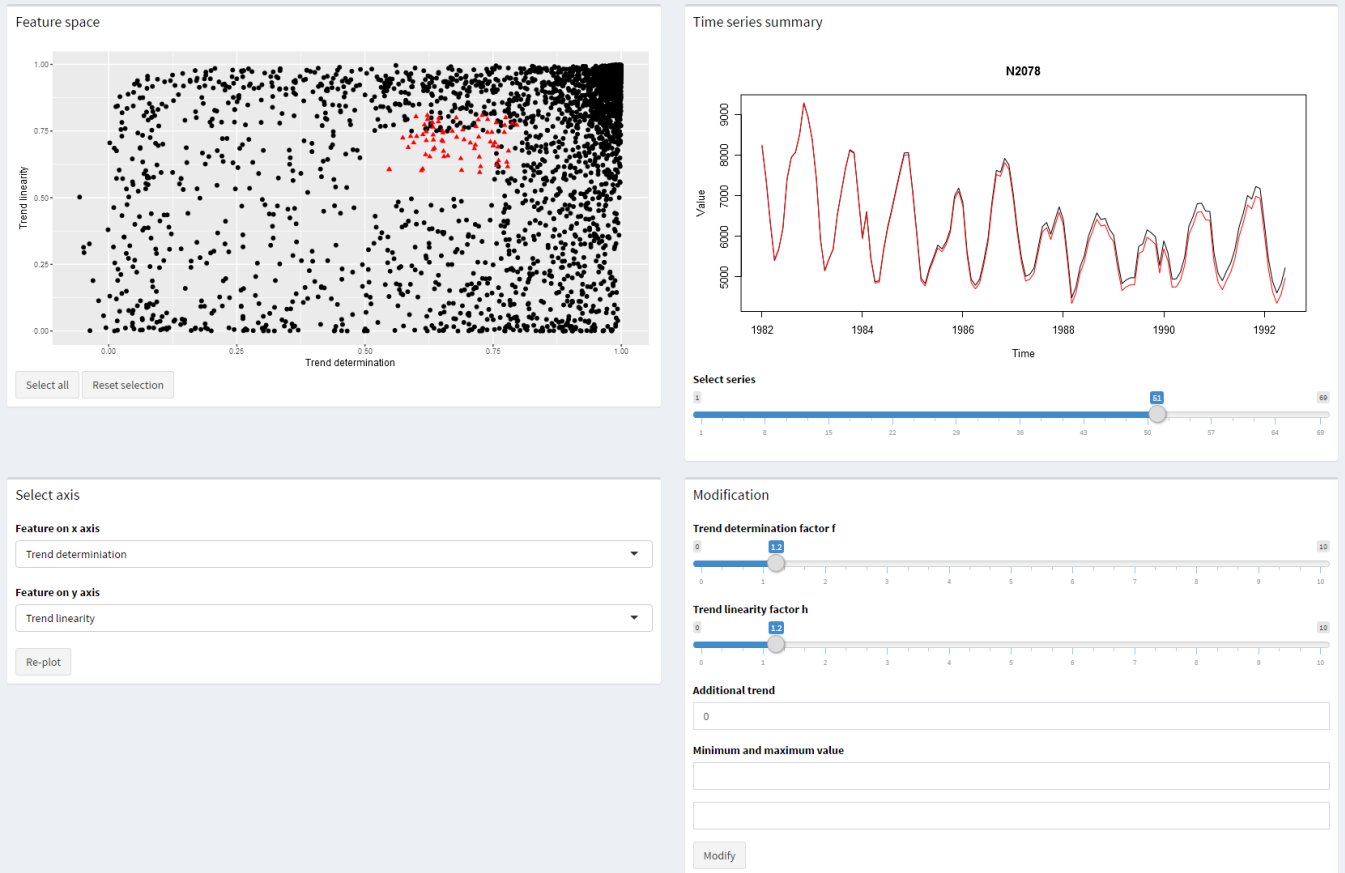


Figure 5: Screenshot of What-If Analysis Tool

4.1 Transformation and Modification

Figure 6 (blue line) shows the aggregate energy consumption of the Smart Metering Project. The consumption is high during winter and low in other seasons. We consider the yearly season as a long-term change because the data set is too short for extracting a cyclical yearly behavior. Therefore, we analyze the season determination feature on the weekly season. By this means, the feature space turns into a histogram. For the modification, we enforce positive consumption values by setting a minimum value of 0, thus, modified time series does not exceed this lower bound. We did not find an expression for factors that guarantees a proportional feature modification, thus, users must keep in mind estimating the factor with respect to the data.

4.2 Example Cases

We give three examples from the Smart Metering Project that model real-world use cases and that show the usage what-if scenarios. As for features, we rely on the season determination representing the weekly season component and the additional trend.

Case 1. We want to assess the risk that arises if the seasonal behavior of all households and SMEs increased. In that case, a utility must take different peaks and lows into account. We express this by a higher season determination and increase this feature by a factor $k = 4$ for the whole data set. A detail of this what-if scenario is shown in Figure 7 (red line). It shows the aggregate consumption of all smart meters and the increased effect on the seasonal behavior compared to the original data (blue line). Also the histogram (Figure 8) shows the increased amount of time series with a high season determination (red bars) compared to the original case (blue bars). The boxplot (Figure 10) summarizes the distribution of season determination and confirms that this feature increases.

Case 2. We pose the question what would happen, if time series that are dominated by their season were less fluctuating. A what-if scenario that expresses this assumption is created by selecting the region $R^2_{seas} \geq 0.5$ in the feature space (Figure 9) and setting a season determination factor $k = 0.5$. The resulting aggregate consumption is shown in Figure 7 (green line), it is the sum of modified series ($R^2_{seas} \geq 0.5$) and unmodified series ($R^2_{seas} < 0.5$). The weekly fluctuation is diminished to a certain extent compared to the original data set but it is still important. The histogram (Figure 8, green bars) shows that the season determination is clearly diminished: there are more time series with a season determination between $0.25 \leq R^2_{seas} \leq 0.50$ and less time series with $0.5 \leq R^2_{seas} \leq 1$. Surprisingly, no modified time series has a season determination less than $R^2_{seas} \leq 0.20$. This underlines that the feature is roughly

Final edited form was published in "SSDBM '17: 29th International Conference on Scientific and Statistical Database Management. Chicago 2017", Art. Nr. 3,
ISBN 978-1-4503-5282-6
https://doi.org/10.1145/3085504.3085507

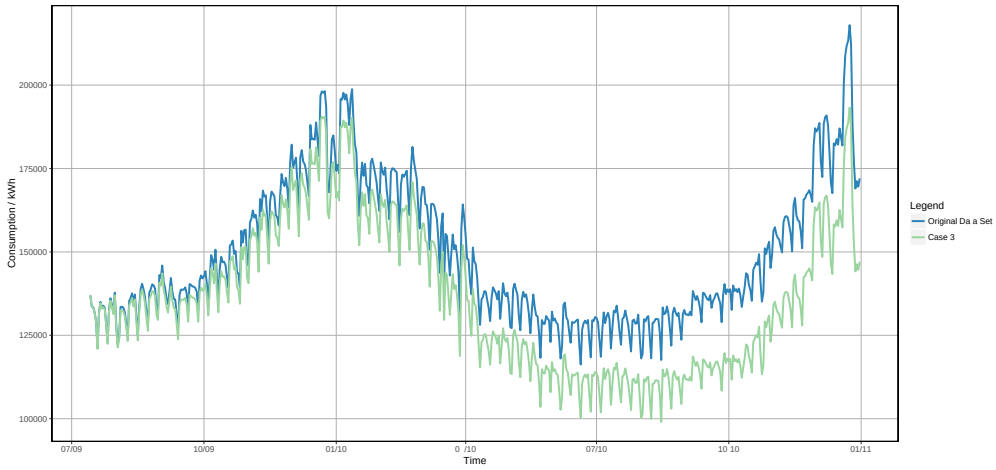


Figure 6: Aggregate Consumption of Smart Metering Project

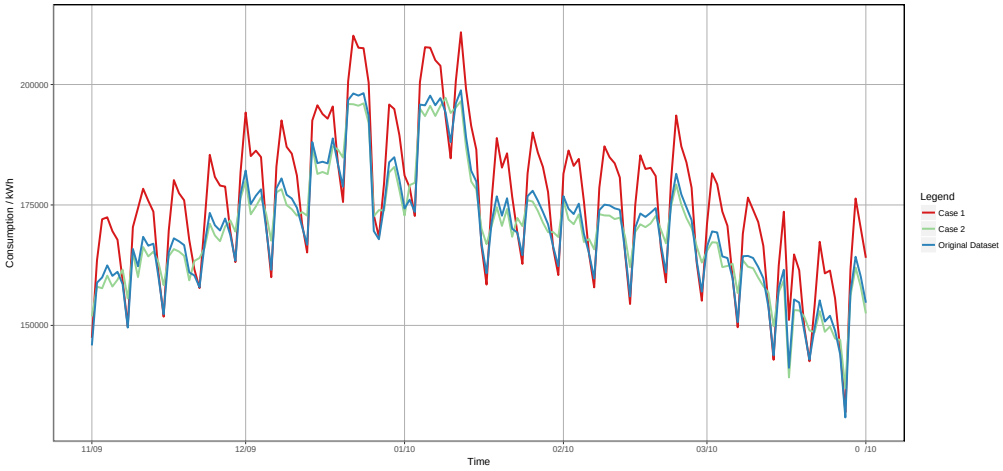


Figure 7: Aggregate Consumption of Smart Metering Project (Detail)

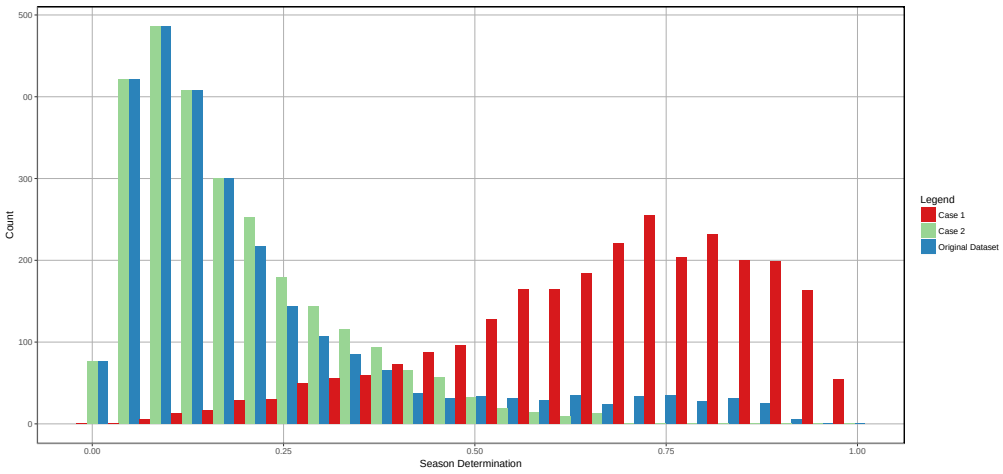


Figure 8: Season Determination of Smart Metering Project

Final edited form was published in "SSDBM '17: 29th International Conference on Scientific and Statistical Database Management. Chicago 2017", Art. Nr. 3,
ISBN 978-1-4503-5282-6
https://doi.org/10.1145/3085504.3085507

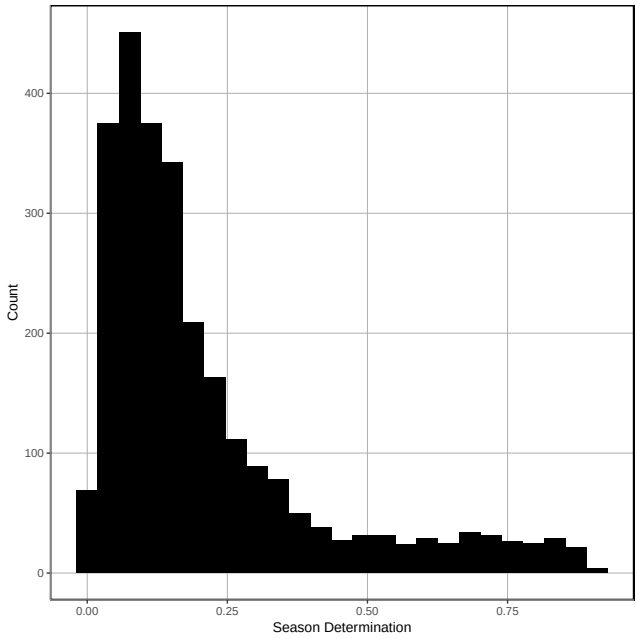


Figure 9: A Histogram as One-Dimensional Feature Space

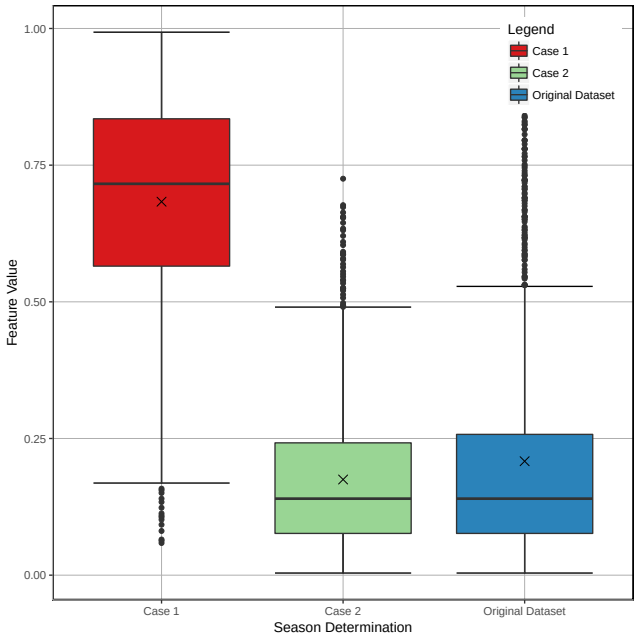


Figure 10: Boxplot of Season Determination

divided by 2, as given by the factor $k = 0.5$. As there are far less time series with a season determination higher than 0.5 than below the overall decrease of season determination is rather small (Figure 10, green bars).

Case 3. We pose the question, what would happen if the consumption of residential households decreased by 20% per year while the consumption of SMEs did not change. It is motivated by the assumption that households diminished their energy consumption due to, for example, more energy-efficient devices, while the consumption of companies did not change. This brings together selection by category attributes, known from OLAP tools, and what-if scenarios with modification, by introducing an additional trend. Since we have a daily granularity within the data and we want a trend decrease by 20% per year, we set $m = -0.2/365$. The resulting plot shows the overall consumption for the original data set and for case 3 (Figure 6). After one year, on July 14, 2010, the consumption is decreased by 12.9% and on December 31, 2010, the difference is already 14.5%, underlining that an important part of consumption is due to private households.

5 DISCUSSION

Our main goal is to present a method that prepares what-if scenarios for time series and that is usable in further analysis tasks. Throughout the steps of transformation, modification and visualization the method returns well-formed data sets that give a substantial insight into scenarios that may arise.

Transforming of time series into components is meaningful in that components and their features are common for a multitude of domains. In our work, we focus on the additive model. Although this assumption is widespread [23], there are other model types too, namely the multiplicative and the mixed additive-multiplicative model. The former may be transformed to the additive model by taking logarithms [22], while the latter includes the additive model as a special case [12, 27]. A check for the best-fitting model would improve the quality of decomposition and should be part of a future extension.

Other time series models may be the basis for the transformation, too, in order to capture other behavior. One could extend our method to multi-seasonal time series which assume more than one season component. Moreover, the method could capture the behavior from, for instance, tenant traces as discussed in [21] which assumes a specific time series model. With these time series models, there are other component features that may be captured and explored by users. In summary, users may adopt the method of what-if scenario generation for an analog use-case.

The application of factors leads to new and useful what-if scenarios, as shown on two example data sets. Trend and season behavior are modifiable by strengthening or weakening a component feature. Most often, the classification by trend and season determination leads to comprehensible feature spaces. Still, there are time series where a slight season results in a high season determination (regarding a negligible residual component), which is mainly due to a very systematic and well decomposed series. In that case, there may be introduced other features that better represent the determination of a season vis-à-vis the trend component.

The proposed visualization covers tasks that are comparable to a query sent against a database. Selecting time series instances corresponds to filtering of a time series with respect to categorical data and features. Previewing the time series corresponds to a projection of the selected subset. Moreover, time series may be aggregated by the sum function if each time series has the same

Final edited form was published in "SSDBM '17: 29th International Conference on Scientific and Statistical Database Management. Chicago 2017", Art. Nr. 3,
ISBN 978-1-4503-5282-6
https://doi.org/10.1145/3085504.3085507

time instances. Finally, the modification of features corresponds to a function applied to the result set.

What-if scenarios may be used in further analysis tools such as forecasting or simulation. Thus, it is a key instrument for business analytics and applies to the predictive as well as to the prescriptive perspective.

6 CONCLUSION AND FUTURE WORK

What-if scenarios play an important role in decision-making in many domains. Applied on time series data, users can get a better insight in recorded data and easily assess different scenarios of the past. Moreover, they contribute to predictive analytics in that users may create forecasts based on what-if scenarios and optimize the outcome with respect to a goal function.

Our aim was to bridge the gap between the hypothetical OLAP queries and spreadsheet-like what-if scenarios. The combination allows users to visually explore data as well as precisely set up new what-if scenarios.

Recently, we presented *Loom*, an application for generating synthetic time series data based on mathematical models and given time series [11]. In the latter case, we reuse time series and their characteristics but we do not modify them. With what-if scenarios, we are now able to systematically cover a feature space and generate data sets that are configurable. Taking the example of the Smart Metering Project, a further query could be "What would be the power consumption of Ireland given an increase of households of 11% by 2020 compared to 2011?" Creating such a scenario includes an assumption on the consumer behaviour as well as the generation of new time series similar to given data. Thus, the combination of these two approaches is important since generated data sets will be flexible in both, data set size and time series features.

Finally, a database system can deeply integrate this method and thus bringing what-if scenarios closer to the data. With this aim, Balmin et al. [2] introduced a first model for an hypothetical scenario on warehouse data and views. The authors define a *select-modify operator* $\hat{\sigma}$ that filters tuples by categorical information and modifies their content. A scenario is a set of ordered hypothetical modifications leading to a new hypothetical data set. With our approach in mind, time series can be also subject to hypothetical queries.

ACKNOWLEDGMENTS

This work is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731232.

REFERENCES

[1] *X-13ARIMA-SEATS Reference Manual V1.1.* http://www.census.gov/ts/x13as/docX13ASHTML.pdf
[2] Andrey Balmin, Thanos Papadimitriou, and Yannis Papakonstantinou. 2000. Hypothetical Queries in an OLAP Environment. In *Proc. of VLDB*. 220 – 231. https://doi.org/10.1109/ICDE.2000.839428
[3] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning. 1990. STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *J. Off. Stat.* 6, 1 (1990), 3–73.
[4] Estella Bee Dagum. 1980. *The X-11-ARIMA seasonal adjustment method.* Statistics Canada.
[5] James R. Evans and Carl H. Lindner. 2012. Business Analytics : The Next Frontier for Decision Sciences. *Decision Line* 43, 2 (2012), 4 – 6.
[6] Ulrike Fischer. 2014. *Forecasting in database systems.* Ph.D. Dissertation. Technische Universität Dresden, Germany. http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-133281
[7] Pierre Gaillard, Yannig Goude, and Raphaël Nedellec. 2016. Additive models and robust aggregation for GEFCom2014 probabilistic electric load and electricity price forecasting. *Int. J. Forecast.* 32, 3 (2016), 1038 – 1050. https://doi.org/10.1016/j.ijforecast.2015.12.001
[8] Maik Günther, Martin Greller, and Mostafa Fallahnejad. 2015. Evaluation of Long-Term Scenarios for Power Generation and District Heating at Stadtwerke München. *Informatik-Spektrum* 38, 2 (2015), 97 – 102.
[9] R. J. Hyndman and G. Athansopoulos. 2013. *Forecasting: principles and practice.* OTexts.
[10] Yanfei Kang, Rob J. Hyndman, and Kate Smith-Miles. 2017. Visualising forecasting algorithm performance using time series instance spaces. *Int. J. Forecast.* 33, 2 (2017), 345 – 358. https://doi.org/10.1016/j.ijforecast.2016.09.004
[11] Lars Kegel, Martin Hahmann, and Wolfgang Lehner. 2016. Template-based Time series generation with Loom. In *Workshops Proc. of EDBT/ICDT.*
[12] M. Kendall and A. Stuart. 1983. *The Advanced Theory of Statistics.* Vol. 3. Griffin, 410 – 414.
[13] Maurice G. Kendall. 1948. *Rank correlation methods.* Griffin.
[14] Spyros Makridakis and Michèle Hibon. 2000. The M3-Competition: results, conclusions and implications. *Int. J. Forecast.* 16, 4 (2000), 451 – 476. https://doi.org/10.1016/S0169-2070(00)00057-1
[15] Oracle. 2003. *SQL for Modeling.* docs.oracle.com
[16] Emanuel Parzen. 1962. On Estimation of a Probability Density Function and Mode. *Ann. Math. Statist.* 33, 3 (1962), 1065–1076.
[17] R Core Team. 2016. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/
[18] Stefano Rizzi. 2009. *What-If Analysis.* Springer US, 3525–3529. https://doi.org/10.1007/978-0-387-39940-9_466
[19] Murray Rosenblatt. 1956. Remarks on Some Nonparametric Estimates of a Density Function. *Ann. Math. Statist.* 27, 3 (1956), 832 – 837.
[20] Fernando Sáenz-Pérez. 2015. Restricted Predicates for Hypothetical Datalog. In *Proc. of PROLE.* 64 – 79.
[21] Jan Schaffner and Tim Januschowski. 2013. Realistic tenant traces for enterprise DBaaS. In *Workshops Proc. of ICDE.* 29 – 35. https://doi.org/10.1109/ICDEW.2013.6547423
[22] Roland Schuhr. 2012. *Prognoserechnung.* Physica-Verlag, Chapter Einführung in die Prognose saisonaler Zeitreihen mithilfe exponentieller Glättungstechniken, 47 – 73.
[23] Robert H. Shumway and David S. Stoffer. 2011. *Time Series Analysis and Its Applications.* Springer. https://doi.org/10.1007/978-1-4419-7865-3
[24] The Commission for Energy Regulation. 2015. CER Smart Metering Project. (2015). www.ucd.ie/issda
[25] Marina Theodosiou. 2011. Forecasting monthly and quarterly time series using STL decomposition. *Int. J. Forecast.* 27, 4 (2011), 1178 – 1195.
[26] Laurynas Šikšnys. 2014. *Towards Prescriptive Analytics in Cyber-Physical Systems.* Ph.D. Dissertation. Aalborg University, Denmark. http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-187556
[27] Xiaozhe Wang, Kate A. Smith, and Rob J. Hyndman. 2006. Characteristic-Based Clustering for Time Series Data. *Data Min. Knowl. Discov.* 13, 3 (2006), 335–364. https://doi.org/10.1007/s10618-005-0039-x