Chapter 2: Theoretical Background

Draft version

This chapter presents a brief introduction to the theory of time series analysis. It is meant to serve as a theoretical foundation of the concepts used in this thesis. Thus, the focus is clearly on those specific concepts, and the chapter should to no means be considered a thorough overview of the complete field of study.

The chapter is divided into five sections. In the first section, a formal definition of time series is given. The second section discusses the main characteristics of time series. Section three describes the different components of time series data and presents methods to split a time series into these components. The fourth section introduces statistical models to forecast time series, in particular the *autoregressive integrated moving average* (ARIMA) class of models. Finally, the fifth section focusses on a specific niche within time series analysis that is used in this thesis, namely time series clustering.

2.1 Time series definition

According to Woodward, Gray, and Elliott (2017), a time series can be defined as follows:

Definition 1 A *time series* is a special type of a stochastic process. A stochastic process $\{Y(t); t \in T\}$ is a collection of random variables, where T is an index set for which all of the random variables are defined on the same sample space. When T represents time, the stochastic process is referred to as a time series. ■

Typically, observations are made at equally spaced time intervals, such as every hour, every day or every year. In such cases, T takes on a discrete set of values, and we refer to them as *discrete-time time series.* On the other hand, *continuous-time time series* arise when T takes on a continuous range of values (Brockwell and Davis 2002). In this thesis, only discrete-time time series are analyzed.

An observed realization of the stochastic process described in Definition 1, is refered to by Woodward, Gray, and Elliott (2017) as a *realization of a time series.* Other works, such as Brockwell and Davis (2002), Shumway and Stoffer (2011) and Hyndman and Athanasopoulos (2018), use the term *time series* for both the data and the stochastic process of which it is a realization. In this thesis, for the sake of simplicity, the latter approach is used, and no notational distinction is made.

Definition 2 A *time series* is a set of observed values $\{y_t\}$ of a stochastic process $\{Y(t); t \in T\}$, where T represents time. ■

From the context it will be clear whether the term time series refers to the process (Definition 1) or its realization (Definition 2). When clarification is needed, it is given locally.

2.2 Time series characteristics

2.2.1 Autocorrelation

Analyzing time series raises unique problems in statistical modeling and inference. In conventional statistics, methods rely on the assumptions of independent and identically distributed random variables. However, in time series analysis, observations made at nearby moments in time are likely to be related. That is, it is likely that there exist internal relationships within a time series. If these relationships are linear, they are called autocorrelation, as defined in Defintion 3 (Shumway and Stoffer 2011).

Defintion 3 *Autocorrelation* measures the linear correlation between two points on the same time series observed at different times. ■

Given a sample $\{y_1, y_2, ..., y_n\}$ of a time series, the degree of dependence in the data can be assesed by computing the *autocorrelation function* (ACF), for each lag h, as defined in Equation 1 (Brockwell and Davis 2002).

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)}, \quad -n < h < n$$

Where $\gamma(h)$ is the autocovariance function of the sample at lag h, as defined in Equation 2.

$$\gamma(h) = n^{-1} \sum_{t=1}^{n-|h|} (y_{t+|h|} - \bar{y})(y_t - \bar{y}), \quad -n < h < n$$

Where n is the length of the sample, y_t is the data value at time t, $y_{t-|h|}$ is the data value at time t minus $|h|$ time periods, and $\bar{y}$ is the mean of the sample. Whenever a time series $\{Y_t\}$ is stationary, a concept that is introduced in the next section, the ACF of the sample $\{y_t\}$ can be used as an estimate for the ACF of $\{Y_t\}$. A time series without autocorrelation, with zero mean and finite variance, is called *white noise.*

Conventional statistical methods can not be appropriately applied to data that exhibit autocorrelation, since the independency assumption is violated. The primary objective of time series analysis, therefore, is to develop mathematical models that are appropriate to use for data with a temporal autocorrelation structure (Shumway and Stoffer 2011). Furthermore, autocorrelation can be used as an advantage, since an internal dependence structure implies that past observations can be used adequately to forecast future values.

2.2.2 Stationarity

In time series analysis, a key role is played by time series whose statistical properties do not vary with time (Brockwell and Davis 2002). Such time series are refered to as *stationary time series.* The most restrictive form of stationarity is defined in Definition 4 (Woodward, Gray, and Elliott 2017).

Definition 4 A time series $\{Y(t); t \in T\}$ is *strictly stationary* if for any $t_1, t_2, ..., t_k \in T$ and any $h \in T$, the joint distribution of $\{Y_{t_1}, Y_{t_2}, ..., Y_{t_k}\}$ is identical to that of $\{Y_{t_{1+h}}, Y_{t_{2+h}}, ..., Y_{t_{k+h}}\}$. ■

However, it is often too hard to mathematically establish the requirement of strict stationarity, since the involved distributions are not known. For most applications, a milder form of stationarity, which only imposes conditions on the first and second moment of a time series, is sufficient. This is officially known as *weak stationarity*, but, in time series analysis, usually just called *stationarity.* It is mathematically defined as in Definition 5 (Woodward, Gray, and Elliott 2017).

Definition 5 A time series $\{Y(t); t \in T\}$ is *stationary* if

1) $E[Y_t] = \mu$ (constant for all t).
2) $Var[Y_t] = \sigma^2$ (constant for all t).
3) $\gamma(t_1, t_2)$ depends only on $t_2 - t_1$. ■

In words, this means that a time series is said to be stationary when the mean and variance are constant over time, and the autocovariance function only depends on the difference between two time points, and not on the time point itself. Note here that each time series that is strictly stationary (Definition 4), is, by definiton, also stationary (Definition 5).

Stationarity is important, since in real-world problems, it is common that only one realization of a time series is available. That means that each random variable in the time series is represented by a single value. This makes it impossible to get an understanding of the underlying probability distributions of those random variables, unless it is assumed that their statistical properties are the same (i.e. the time series is stationary). In that case, the statistical properties of the whole sample can be used to estimate the statistical properties of each individual probability distribution. An understanding of the underlying propbability distributions, in turn, is especially important when the goal is to forecast how the time series will behave at future time points. When the statistical properties of the time series have been constant over time in the past, one can simply predict that they will remain constant in the future.

In most practical applications, non-stationary time series are the rule rather than the exception. Luckily, by applying mathematical transformations, it often possible to render a non-stationary time series as, approximately, stationary. This process is refered to as *stationarizing* a time series (Nau 2018). Stationarizing is used a lot in statistical forecasting methods, which are discussed in section 2.4.

2.2.3 Spectral entropy

Text

2.3 Time series components

2.3.1 Definitions

A time series can consist of various underlying patterns. Each of those patterns is considered a distinct component of the time series, with its own properties and behaviour. Splitting a time series into its components is known as *time series decomposition*. It enables a seperate analysis of all the components, which helps to better understand the dynamics of a time series, but can also be useful in forecasting, as is showed later in this chapter.

Hyndman and Athanasopoulos (2018) define three main components of a time series: a trend-cycle component, a seasonal component and a remainder component. For simplicity, the trend-cycle component is usually called just the trend component, which is done in this thesis as well.

Defintion 4 The *trend component* is the combination of the trend and cyclical pattern of a time series. A trend exists when there is a long-term, not necessarily linear, increase or decrease in the data. A cyclical pattern occurs when the data exhibit rises and falls that are not of a fixed frequency. ■

Defintion 5 The *seasonal component* contains the seasonal pattern of a time series. A seasonal pattern occurs when a time series is affected by seasonal factors such as the time of the year or the day of the week. Seasonality is always of a fixed and known frequency. ■

Defintion 6 The *remainder component* is the remaining variation in a time series after the trend and seasonal components are removed. ■

There exist several different methods for the decomposition of a time series. Most of them are based on the classical decomposition method, which is discussed in the next section. A more sophisticated approach is known as STL, and is covered in section 2.2.2.3.

2.3.2 Classical decomposition

The oldest and simplest method for the decomposition of a time series is refered to as classical decomposition by Hyndman and Athanasopoulos (2018). They present a stepwise approach for the use of the method, which is summarized in this section. Classical decomposition can be applied in two different forms. In the additive form, a time series is assumed to be the sum of its components, as shown in Equation 2.

$$y_t = T_t + S_t + R_t$$

In the multiplicative form, a time series is assumed to be the product of its components, as shown in Equation 3.

$$y_t = T_t \times S_t \times R_t$$

Where, for both Equation 2 and Equation 3, y_t is the data, T_t is the trend component, S_t is the seasonal component and R_t is the remainder component.

Additive decomposition is the appropriate form when the amplitude of the variation around the trend is relatively constant. On the other hand, when the amplitude of the variation around the trend changes with the level of the trend, multiplicative decomposition should be used.

In both the additive and multiplicative form of classical decomposition, the first step is to estimate the trend component. This is done by smoothing the data with a symmetric moving average filter of order m, where m is a non-negative integer. That is, the estimate of the trend component at time t is the average of all the data values within a window of m time periods centered at t, as shown in Equation 4. Usually, m is set to be equal to the seasonal period of the time series, which, in turn, is the number of observations per seasonal cycle. For example, when working with daily data that show a weekly seasonal pattern, the seasonal period is 7.

$$\hat{T}_t = \frac{1}{m} \sum_{j=-k}^{k} y_{t+j}$$

Where $k = (m-1)/2$. The detrended time series data are then calculated by removing the estimated trend component from the original data. In the case of additive decomposition by substraction, $y_t - \hat{T}_t$, and in the case of multiplicative decomposition by division, $y_t/\hat{T}_t$.

The seasonal component is estimated by averaging the detrended data values per season, as shown in Equation 5. Using again the example of daily data with a weekly seasonal pattern, that would mean that the estimated seasonal component for a specific Monday is the average value of all Monday observations in the dataset, the estimated seasonal component for a specific Tuesday is the average value of all Tuesday observations in the dataset, and so on.

$$\hat{S}_t = \frac{1}{n_t} \sum_{i=1}^{n_t} (\omega_t)_i$$

Where ω_t is a vector containing all the detrended values belonging to the same season as y_t, and n_t is the length of ω_t. Usually, the estimated seasonal component values for each season are adjusted such that they add up to 0 in the case of additive decomposition and 1 in the case of multiplicative decomposition.

Finally, the remainder component is estimated by removing both the estimated trend component and the estimated seasonal component from the original time series. For additive decomposition, this is done by applying Equation 6.

$$\hat{R}_t = y_t - \hat{T}_t - \hat{S}_t$$

For multiplicative decomposition, Equation 7 is used.

$$\hat{R}_t = \frac{y_t}{\hat{T}_t \hat{S}_t}$$

Classical decomposition is generally praised for its simplicity, but has several disadvantages compared to some of the more modern decomposition methods (Hyndman and Athanasopoulos 2018). As a consequence of the use of a symmetric moving average filter, there are no trend component estimates available for the first few and last few observations of the time series. Therefore, also the remainder component estimate lacks these values. This is mainly problematic when forecasting, as is showed later in this chapter. Furthermore, the seasonal component stays constant over all the seasonal cycles, and cannot capture slight changes over time. Especially when working with longer time series, this may be an inappropriate representation of the truth. Finally, classical decomposition is not robust to extreme values in the data.

2.3.3 STL decomposition

A widely used method that is based on classical decomposition, but deals with many of the limitations mentioned above, is known as STL. It stands for *A Seasonal-Trend decomposition procedure based on Loess*, and was developed by R. B. Cleveland et al. (1990). In this section, their methodology is summarized. STL estimates all three components for every observation in a time series, and can also handle missing values in the data. Both the trend and seasonal component are robust and not distorted by extreme values. Furthermore, the seasonal component is not fixed, but can vary slightly over time.

As its name already implies, STL is based on loess, also known as locally-weighted regression. Loess was developed by Cleveland and Devlin (1988), and is a non-parametric regression technique, often used for smoothing, that fits weighted least squares regression curves to local subsets of a dataset. Joining them together forms the loess regression curve $\hat{g}(x)$. More specifically, for each value of x, $\hat{g}(x)$ is computed in the following way. First, a positive integer q is chosen, which defines the neighborhood width. That is, the q observations that are closest to x are selected as neighbors of x. Each of these observations is given a weight based on its distance to x, in a way that the closest observations get the highest weight. Let W be the tricube weight function as defined in Equation 8.

$$W(u) = \begin{cases} (1-u^3)^3 & 0 \leq u < 1 \\ 0 & u \geq 1 \end{cases}$$

Then, the neigborhood weight for each observation x_i is calculated with Equation 9.

$$\upsilon_i = W\left(\frac{|x_i - x|}{\lambda_q(x)}\right)$$

Where $\lambda_q(x)$ is the distance of the qth farthest observation from x. Then, $\hat{g}(x)$ is calculated by fitting a polynomial regression of degree d to x, using weighted least squares with the neigborhood weights υ_i. Usually, d is either 1 or 2, corresponding respectively to a locally-linear regression and a locally-quadratic regression. Since the loess regression curve is smooth, there is no need to compute $\hat{g}(x)$ at all possible values of x. In general, the computation of $\hat{g}(x)$ as described above is only performed at a finite set of locations, and interpolated elsewhere.

STL uses loess for several smoothing operations, that, when performed on a time series, lead to estimations of the trend, seasonal and remainder components of the data. The method is build up of two loops: an inner loop nested inside an outer loop. In the inner loop, the estimates of the seasonal and trend component are updated once, in a stepwise manner, which is described below.

Step 1. The inner loop starts with computing the detrended time series data $y_t - \hat{T}_t$ from the original time series data y_t. In the initial pass through the inner loop, there is no estimation of T_t yet, and $\hat{T}_t$ is set equivalent to 0. That is, it is assumed there is no trend at all. This may be a rather poor estimate, but inside the loop, it will soon be updated to something more reasonable. In all successive passes through the loop, the estimated trend component that resulted from the previous loop is used.

Step 2. In the second step, the detrended time series is split up into subsets, with each subset containing all the data belonging to one specific season. That is, there will be n_p different subsets, where n_p is the number of observations per seasonal cycle. Each of those subsets is smoothed by loess, with $q = n_s$ and $d = 1$. n_s is refered to as the seasonal smoothing parameter and its value must be chosen by the analyst. It basically determines how much the seasonal component is allowed to change over time. High values of n_s allow little variation, while low values can lead to overfitting. The smoothed values of all the subsets are then binded back together into a temporary seasonal component C_t. Each end of C_t is extended n_p positions, such that C_t has $2 \times n_p$ observations more than the original time series.

Step 3. In the third step, any trend that may have contaminated C_t is identified. This is done by applying a sequence of smoothers, called a low-pass filter, to C_t. It starts with a moving average of length n_p, followed by another moving average of length n_p, followed by a moving average of length 3, followed by a loess smoothing with $q = n_l$ and $d = 1$. Just as earlier with n_s, the low-pass filter smoothing parameter n_l should be chosen by the analyst. The output of the third step is called L_t. Since moving averages are used, the first n_p observations and the last n_p observations of C_t will not have a smoothed value in L_t. However, this was already accounted for by extending C_t in step 2. That is, L_t is again of the same length as the original time series.

Step 4. In the fourth step, the seasonal component is estimated by detrending the temporary seasonal component. That is, $\hat{S}_t = C_t - L_t$.

Step 5. In the fifth step, the deseasonalized time series data $y_t - \hat{S}_t$ are computed from the original time series data y_t.

Step 6. In the sixth and last step of the inner loop, the estimation of the trend component, $\hat{T}_t$, is calculated by loess smoothing the deseasonalized time series with $q = n_t$ and $d = 1$. The trend smoothing parameter n_t should be chosen by the analyst.

The outer loop of STL starts with n_i iterations of the inner loop. The estimations of the trend and seasonal components that follow from the passes through the inner loop, are used to estimate the remainder component with Equation 10.

$$\hat{R}_t = y_t - \hat{T}_t - \hat{S}_t$$

For each observation in the time series, a robustness weight is calculated. This weight reflects how extreme the value of the remainder component of that observation is, in a way that a very extreme value is given a very low, or even zero, weight. Let B be the bisquare weight function as defined in Equation 11.

$$B(u) = \begin{cases} (1-u^2)^2 & 0 \leq u < 1 \\ 0 & u \geq 1 \end{cases}$$

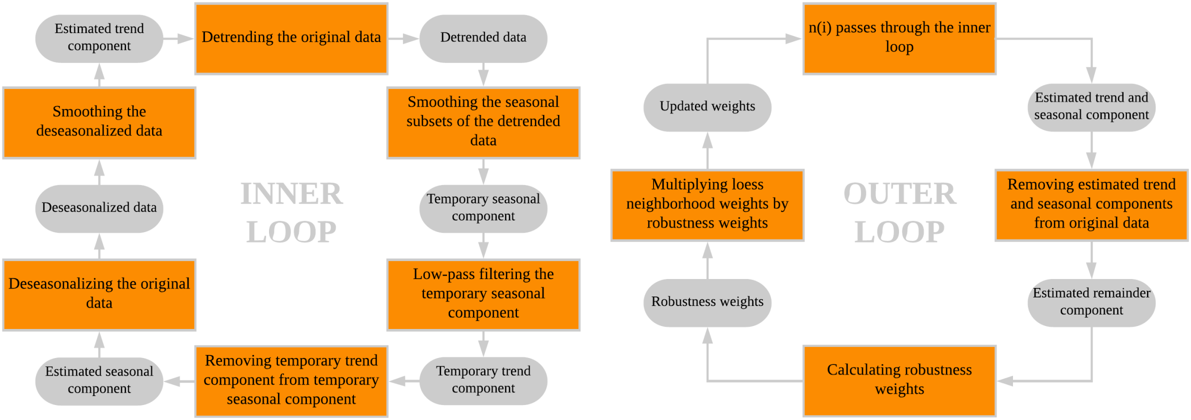
Then, the robustness weight at time point t is calculated with Equation 12.

$$\rho_t = B\left(\frac{|R_t|}{6median(|R_t|)}\right)$$

After the first pass of the outer loop, the next iteration starts again with n_i passes through the inner loop. However, in the loess smoothing in step 2 and step 6, each neighborhood weight v_t is now multiplied by its corresponding robustness weight ρ_t, such that extreme values have less influence on the estimates of the trend and seasonal components. Also, the estimated trend component that resulted from the last inner loop in the previous outer loop, is now used as first value of $\hat{T}_t$, rather than 0. In total, the outer loop is carried out n_o times.

STL is designed for additive decomposition. However, a multiplicative version can be obtained by first log transforming the data, and finally back-transforming the components. This is based on the logarithm product rule, which states that $log(a) + log(b)$ is equivalent to $log(a \times b)$ (Hyndman and Athanasopoulos 2018).

The complete methodology of STL as described above is summarized in Figure 1.


Estimated trend component
Detrending the original data
Detrended data
Smoothing the deseasonalized data
Smoothing the seasonal subsets of the detrended data
INNER LOOP
Deseasonalized data
Temporary seasonal component
Deseasonalizing the original data
Low-pass filtering the temporary seasonal component
Estimated seasonal component
Removing temporary trend component from temporary seasonal component
Temporary trend component
n(i) passes through the inner loop
Updated weights
Estimated trend and seasonal component
Multiplying loess neighborhood weights by robustness weights
OUTER LOOP
Removing estimated trend and seasonal components from original data
Robustness weights
Estimated remainder component
Calculating robustness weights

2.4 Time series forecasting

2.4.1 Forecasting models

Often, the main aim of time series analysis is to forecast future values of a time series. In some cases, this can be done by using external exploratory variables. One could for example try to forecast the profit of ice cream sales by using air temperature as an exploratory variable in a linear regression model. However, there are several reasons not to forecast time series in this way, as summed up by Hyndman and Athanasopoulos (2018). Firstly, the underlying system of the forecasted time series may not be sufficiently understood, and even if it is, the relations with exploratory variables may be too complex. Secondly, when forecasting future values of a time series, also the future values of the exploratory variables should be known, which means that each exploratory variable should be forecasted seperately before the response variable can be forecasted. This may be too difficult to do accurately, and even when it is possible, it remains a very time consuming task. Especially when the only aim is to know what will happen, and not why it will happen, it is not worth the effort. Finally, modeling a time series with convential statistical method like linear regression will likely result in model errors that exhibit autocorrelation, which implies that such models are not able to capture all the dynamics of the data. Thus, produced forecast are not as efficient, and, probably, not as accurate as can be.

Instead, in time series analysis, the internal dependence structure of a time series is used to forecast future values as a function of the current and past values (Shumway and Stoffer 2011). Obviously, this primarily requires a good understanding of that structure, which is obtained by describing the process that generated the data with a time series model, as defined in Defintion 7, adapted from Brockwell and Davis (2002).

Definition 7 A *time series model* for an observed realization $\{y_t\}$ of a time series $\{Y_t\}$ is a specification of the joint distributions, or possibly only the means, variances and covariances, of the random variables that $\{Y_t\}$ comprises. ■

One of the most famous and widely used group of time series models is known as the *autoregressive integrated moving average* (ARIMA) models, developed by Box and Jenkins (1970). In this thesis, ARIMA is used as well. The next section gives a summary of its theory, based on Brockwell and Davis (2002), Chapter 5, Shumway and Stoffer (2011), Chapter 3, Hyndman and Athanasopoulos (2018), Chapter 8, and Hyndman and Khandakar (2008).

2.4.2 ARIMA

2.4.2.1 Structure

An ARIMA model is a combination of an *autoregressive* (AR) and *moving average* (MA) model, preceded by a differencing operation on the original data. An autoregressive model of order p, commonly refered to as an AR(p) model, is based on the assumption that the current value of a time series is a linear combination of p previous values, as showed in Equation 13.

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \epsilon_t$$

Where y_t is the current value of the time series at time period t, ϵ_t is the random error (i.e. white noise) at time t, $\phi_1, ..., \phi_p$ are model parameters.

A moving average model of order q, commonly refered to as an MA(q) model, is based on the assumption that the current value of a time series is a linear combination of q previous errors, as showed in Equation 14.

$$y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q}$$

Where y_t is the current value of the time series at time period t, ϵ_t is the error at time period t, which is assumed to be white noise, $\theta_1, ..., \theta_q$ are model parameters.

AR(p) and MA(q) models can be combined into an autoregressive moving average model of order (p, q), commonly refered to as ARMA(p, q). That is, in such a model, the current value of a time series is a linear combination of both p previous values and q previous errors, as showed in Equation 15.

$$y_t = \phi_1 y_{t-1} + ... + \phi_p y_{t-p} + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q} + \epsilon_t$$

ARMA(p, q) models require the forecasted time series to be stationary. When working with non-stationary time series, it is often possible to stationarize the series by differencing it one or more times. The first order difference of a time series is the series of changes from one time period to the next, as shown in Equation 16.

$$\nabla y_t = y_t - y_{t-1}$$

Where ∇y_t is the first order difference of y_t. When the first order difference is still non-stationary, the second order difference $\nabla^2 y_t$ can be computed by taking again the first order difference of ∇y_t, and so on. The original non-stationary time series that needed to be differenced in order to get stationary, is called an *integrated* version of the stationary series. That is why a model that first stationarizes the data by applying a d-th order difference, before fitting an ARMA(p, q) model, is called an autoregressive integrated moving average model of order (p, d, q), commonly refered to as ARIMA(p, d, q). That is, in such a model, the current value of the $d-$th order difference of a time series is a linear combination of both p previous values and q previous errors, as showed in Equation 17.

$$\nabla^d y_t = \phi_1 \nabla^d y_{t-1} + ... + \phi_p \nabla^d y_{t-p} + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q} + \epsilon_t$$

Where $\nabla^d y_t$ is the d-th order difference of y_t. Note here that ARIMA(p, d, q) is a general form of all the other models discussed earlier in this section. For example, an AR(1) model can also be written as ARIMA(1,0,0), an ARMA(2,1) as ARIMA(1,0,2), and so on.

The process of finding an appropriate ARIMA(p, d, q) model that represents a time series is known as the Box-Jenkins modelling procedure and consists of three stages, named model selection, parameter estimation and model checking. All these stages are described seperately in the next three subsections.

2.4.2.2 Model selection

In the model selection stage, p, d and q are chosen. In this process, d is selected first, such that the choice of p and q will be based on a stationary time series. An appropriate value for d can be found by inspecting the plotted data y_t, with time on the x-axis, and define visually if the data are stationary. If not, then difference the data once, and inspect the plot of ∇y_t. If ∇y_t does not seem stationary either, take the second-order difference $\nabla^2 y_t$, and so on. In general, however, it is not recommended to difference more than two times. As an addition to the time plots, plotting the sample autocorrelation function of the data can help to identify stationarity. Non-stationary data show a slow decay in autocorrelation as the time lag increases, while for stationary data, the autocorrelation will drop to zero relatively fast.

Once d has been set, either p or q can be selected by inspecting the autocorrelation function plot and the partial autocorrelation function plot of the differenced data. The partial autocorrelation function (PACF) is the relationship between an observation at time t and and observation at time $t-k$, removing the effects of all time lags inbetween, i.e. $1, 2, ..., k-1$. Then, appropriate values for either p or q are found with the following rules of thumb:

- The PACF plot of an ARIMA(p,d,0) process cuts of after lag p, and the ACF plot tails off.
- The ACF plot of an ARIMA(0,d,q) process cuts of after lag q, and the PACF plot tails off.

When dealing with ARIMA(p, d, q) processes where both $p > 0$ and $q > 0$, the ACF plot and PACF plot will both tail off, and finding appropriate values for p and q turns into a trial-and-error approach, where models with different combinations of p and q are compared.

The methodology as described above is used often, but involves a lot of manual interventions. This makes it a rather subjective way of working, that is labor intensive, especially when a large number of time series needs to be modelled, and requires expert knowledge. Therefore, several automated approaches to select p, d and q have been proposed. One of them is the Hyndman-Khandakar algorithm, which methodology is summarized below, in a simplified way. For the full details, see Hyndman and Khandakar (2008).

Step 1. To define d, the Hyndman-Khandakar algorithm uses the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test, which is a statistical test used to determine stationarity of a time series. Only if there is enough statistical evidence, the null hypothesis that the time series is stationary will be rejected, and the time series is instead considered to be non-stationary. The detailed mathematics underlying the test can be found in Kwiatkowski et al. (1992).

Using the KPSS test, first, the original data y_t are tested for stationarity. When y_t are considered stationary, $d = 0$, and when considered non-stationary, the first-order differenced data ∇y_t are tested for stationarity. Again, when ∇y_t are considered stationary, $d = 1$, and when considered non-stationary, the second-order differenced data $\nabla^2 y_t$ are tested for stationarity. This process is repeated until a stationary series is obtained.

Step 2. In the second step, four different models are fitted to the d-times differenced data.

- An ARIMA(0, d, 0) model.
- An ARIMA(1, d, 0) model.
- An ARIMA(0, d, 1) model.
- An ARIMA(2, d, 2) model.

Then, the model with the lowest AIC is selected. AIC, which stands for Aikake's Information Criterion, is a measure for the goodness-of-fit of a model, and can be calculated with Equation 2.x.

$$AIC = -2\log(L) + 2k$$

Where L is the likelihood of the data, and k is the number of free parameters in the model. In this case, $k = p + q + l + 1$, where $l = 1$ when a non-zero constant is included, and $l = 0$ otherwise. The '+1' term is included, since the variance of the residuals is also a parameter. To find the best fitting model, AIC should be minimized. The idea behind AIC is the following. The likelihood monotonically increases when more parameters are added to the model, and therefore, only maximizing the likelihood would favor a model that overfits the data. AIC prevents such overfiting, by penalizing the likelihood with a term that is proportional to the number of parameters used in the model.

Step 3. In the third step, several variations of the model that was selected in step 2, are fitted to the d-times differenced data. These variations include:

- Models where either p or q vary ±1 from the selected model, given that $p, q \ngtr 5$.
- Models where both p and q vary ±1 from the selected model, given that $p, q \ngtr 5$.

From the selected model and all its variations, the model with the lowest AIC is chosen to be the new selected model, and step 3 is repeated. The algorithm stops when there are no variations of the selected model that have a lower AIC. In that case, the selected model is the optimal model, and forms the output of the Hyndman-Khandakar algorithm. The complete methodology of the algorithm as described above is summarized in Figure x.

2.4.2.3 Parameter estimation

When p, d and q are defined, the model parameters $\phi_1, ..., \phi_p$ and $\theta_1, ..., \theta_q$ need to be estimated. Usually, this is done with *maximum likelihood estimation* (MLE). The likelihood is the probability of obtaining the observed data, given the model and specific parameter values. The parameter values that maximize the likelihood are called the maximum likelihood estimators of the true parameters, and will be used as the parameter estimates in the fitted ARIMA(p,d,q) model, which then is refered to as the maximum likelihood ARIMA(p,d,q) model. The detailed mathematical description of MLE for ARIMA models can be found in Brockwell and Davis (2002), section 5.2.

Note here that the Hyndman-Khandakar algorithm already produces a fitted model as output, and the parameter estimation as described in this section is done inside the algorithm, each time a model is fitted to the d-times differenced data.

2.4.2.4 Model checking

Model checking involves identifying if the fitted model is adequate. This is done by inspecting its residuals, which are defined as the difference between the actual observations and the corresponding fitted values, as shown in Equation 2.x.

$$e_t = y_t - \hat{y_t}$$

If the maximum likelihood ARIMA(p,d,q) model is the true process that generated the data, the residuals should be completely white noise. Recall, however, that the model is an estimation of the true process. Therefore, a good model that fits the data well, should have residuals with properties that *approximately* reflect those of white noise, i.e. a zero mean and no autocorrelation. If autocorrelation is present in the residuals, this means that there is still information left in the data, which could be used to create more accurate forecasts. A non-zero mean will lead to biased forecasts.

Autocorrelation in the residuals can be detected by a visual interpretation of the residual ACF plot, which will always show some autocorrelation, due to random variation. Therefore, given that n is the length of the modelled time series, and assuming a normal distribution, the residuals are considered to be uncorrelated when for at least 95% of the time lags, the residual ACF lies within the interval $[-1.96/\sqrt{n}, 1.96/\sqrt{n}]$. Instead, the Ljung-Box test can be used, which tests the null hypothesis that the residuals are independent, against the alternative that they are not. The detailed mathematics underlying the test can be found in Ljung and Box (1978).

Usually, several computations within the model fitting and forecasting process build on the assumption that the data come from a normally distributed population. For example, in MLE and the calculation of AIC, Gaussian likelihood is commonly used. Furthermore, prediction intervals of forecasts are in general derived from the normal distribution. Normally distributed residuals indicate that these assumptions were valid, and are therefore a valuable property of a model. However, as stated by Brockwell and Davis (2002), using Gaussian likelihood is sensible even when the data are not normally distributed.

2.4.2.5 Forecasting

A fitted ARIMA(p, d, q) model can then be used to forecast the future values of a time series. To do so, Equation 2.x is rewritten, such that the current value of the time series, y_t, is replaced by a future value of the time series, y_{t+h}, as shown in Equation 2.x.

$$\nabla^d y_{t+h} = \hat{\phi}_1 \nabla^d y_{t+h-1} + ... + \hat{\phi}_p \nabla^d y_{t+h-p} + \hat{\theta}_1 \epsilon_{t+h-1} + ... + \hat{\theta}_q \epsilon_{t+h-q} + \epsilon_{t+1}$$

Where h is the forecast horizon, i.e. the number of time lags ahead at which the forecast is made, p, d and q are known and constant, and $\hat{\phi}_1, ..., \hat{\phi}_p$ and $\hat{\theta}_1, ..., \hat{\theta}_q$ are the estimated parameter values, which are also constant.

When $h > 1$, more than one forecast has to be made. For example, the forecast of $\nabla^d y_{t+2}$, the value of the time series two time lags ahead, is based on $\nabla^d y_{t+2-1}$, the value of the time series one time lag ahead. Therefore, $\nabla^d y_{t+2-1}$ needs to be forecasted first, before $\nabla^d y_{t+2}$ can be forecasted. In general, this means that the uncertainty of the forecasts increases as h increases. This uncertainty is expressed by means of a prediction interval. Most often, the 95% prediction interval is use. Assuming normally distributed residual, the lower and upper bound of the 95% prediction interval for the h-step forecast can be calculated by Equation 2.x and 2.x, respectively.

$$\ell = \hat{y}_{t+h} - 1.96\hat{\sigma}_h$$
$$\upsilon = \hat{y}_{t+h} + 1.96\hat{\sigma}_h$$

Where ℓ is the lower bound of the 95% prediction interval, υ is the upper bound of the 95% prediction interval, $\hat{y}_{t+h}$ is the forecasted value h time lags ahead. $\hat{\sigma}_h$ is the estimated standard deviation of the forecast distribution h time lags ahead, which is explained below. The 95% prediction interval can be interpreted as follows: there is a 95% probability that $\ell \leq y_{t+h} \leq \upsilon$.

Recall that in Defintion 1, a time series was defined as a collection of random variables. In fact, to state it statistically correct, it is the distribution of the random variable h time lags ahead that is forecasted, rather than an individual value. This distribution is refered to as the forecast distribution, and the forecasted value, also known as the point forecast, is then taken to be the mean of the forecast distribution. In Equation 2.x and 2.x, $\hat{\sigma}_h$ is the estimated standard deviation of the forecasted distribution, assuming it is a normal distribution with mean y_{t+h} and variance σ_h^2. When $h = 1$, the residual standard deviation σ_ϵ is a good estimate for σ_h. However, for $h > 1$, computation get more complex. For a detailed description, see Shumway and Stoffer (2011), section 3.5.

2.4.2.6 Accuracy evaluation

2.4.2.7 Transformations

2.4.3 Naïve forecasts

2.4.4 Seasonality

2.5 Time series clustering

2.5.1 Dissimilarity measures

Time series clustering is a specific domain within the field of time series analysis. Given a set of individual time series, its objective is to group similar time series into the same cluster (Keogh and Lin 2005). Logically, this involves the calculation of a measure that represents the similarity, or dissimilarity, between two time series. Whenever the time series are of the same length, and observed at the same timestamps, a simple dissimilarity measure can be calculated by summing the ordered point-to-point distances between them. The most used distance function, in this sense, is the Euclidean distance, as defined in Equation 2.x (Cassisi et al. 2012).

$$d(Y,X) = \sqrt{\sum_{t=1}^{n}(y_t - x_t)^2}$$

Where $d(X,Y)$ is the dissimilarity value between time series Y and X, y_t and x_t are the observations in respectively Y and X at time t, and n is the length of Y and X. A higher value of $d(X,Y)$ implies less similar time series. By definition, $d(X,Y) \geq 0$.

Using Euclidean distance as a dissimilarity measure is simple, but has drawbacks for certain applications. For example, it can not handle time series of different length, it is sensitive to outliers, and it can not capture out-of-phase similarities, which occur when two time series show similar patterns, but shifted over time. To deal with one or more of these drawbacks, several other dissimilarity measures for time series were developed, of which the *dynamic time warping distance* is best known. Dynamic time warping is based on classical algorithms for comparing discrete sequences with continuous sequences, and basically replaces the one-to-one comparison of Euclidean distances with a many-to-one comparison. However, despite its simplicity, the Euclidean distance approach still turns out to be very competitive with the more complex methods (Cassisi et al. 2012).

Once the dissimilarity values for all possible combinations between the analyzed time series are calculated, they are stored in an $n \times n$ matrix, where n is the number of analyzed time series, and the diagonal entries equal zero. Such a matrix is commonly refered to as a dissimilarity matrix, and can be used to cluster the time series with conventional clustering algorithms. In time series analysis, k-means clustering and hierarchical clustering are the most popular options for this task (Wang, Smith, and Hyndman 2006). The latter is used in this thesis, and its theory is summarized briefly in the next sub-section, based on Gan, Ma, and Wu (2007), Chapter 7 and Chapter 17.

2.5.2 Hierarchical clustering

Hierarchical clustering algorithms divide a set of data points into a sequence of nested partitions, where each partition consists of a different number of clusters. Two main types are distinguished: *agglomerative* hierarchical clustering, and *divisive* hierarchical clustering. In the first, the starting point is a partition in which all data points form a cluster on their own. Then, the two closest clusters, based on a pre-defined dissimilarity measure, are merged. This process is repeated until all data points are in one single cluster. The divisive method works the other way around: all data points start in the same cluster, which is split repeatedly, until the number of clusters equals the number of data points. Agglomerative hierarchical clustering is the most popular of the two types, and forms the focus of this section.

Again, defining a suitable dissimilarity measure is a core task in the process. The simplest way of doing so, is known as the *single link method*, where the dissimilarity value between two clusters is defined as the shortest possible distance from a data point in the first cluster, to a data point in the second cluster. In contradiction, the *complete link method* calculates the dissimilarity value as being the longest possible distance from a data point in the first cluster, to a data point in the second cluster. Other methods include the *group average method*, which calculates the average of the shortest distances between all possible pairs of data points, and the *centroid method*, which calculates the shortest distance between the cluster centroids. In all cases, the Euclidean distance is commonly used as the distance function.

A more general approach is known as the *Ward method*, developed by Ward Jr. (1963). In this method, the dissimilarity between two clusters is defined as the loss of information when the clusters are merged. The information loss of a merge between two clusters is quantified with Equation 2.x.

$$\Delta I = I(C_m) - I(C_1) - I(C_2)$$

Where ΔI is the information loss, C_m is the merge between clusters C_1 and C_2, and $I(C_m)$, $I(C_1)$ and $I(C_2)$ are the information criteria of respectively C_m, C_1 and C_2. Ward Jr. (1963) did not put a hard restriction on how such an information criterion should be quantified, but usually, it is set to be the *error sum of squares* of the cluster, calculated with Equation 2.x.

$$I(C) = \sum_{i=1}^{n} (c_i - \mu(C))^2$$

Where c_i are the data points in C, $\mu(C)$ is the center of mass of C, and n is the number of data points in C. In other words, $I(C)$ is the sum of the Euclidean distances from all data points in the cluster, to the center of mass of the cluster.

Since a hierarchical clustering algorithm produces a sequence of partitions, it is not needed to define the desired number of clusters, k, in advance. However, if one is interested in obtaining only one single partition, it is necessary to find a suitable value of k. It is common practice to do so by visually interpreting the dendrogram of the hierarchical clustering, which is a diagram representing the output in a tree structure, but automated approaches have been developed as well. Most of them are based on the idea that in an ideal situation, clusters are compact and clearly seperated from each other. However, minimizing the variance within the clusters will always favor the situation where each data point forms a cluster on its own, while maximizing the variance between the clusters will always favor the situation where all data points are together in one cluster. Therefore, most approaches combine those two operations, in order to find the best possible partition.

An example of such an approach is the *Dunn Index*, developed by Dunn (1974). For a specific partition into k clusters, it calculates the ratio of the smallest distance between two data points that are not in the same cluster to the largest distance between two data points that are in the same cluster. This is shown in Equation 2.x.

$$V(\Lambda_k) = \min \left\{ \min \left(\frac{D(C_i, C_j)}{\max diam(C_l)} \right) \right\}$$

Where $V(\Lambda_k)$ is the Dunn Index of a partition Λ_k with k clusters, $1 \leq i, j, l \leq k$, $D(C_i, C_j)$ is the Euclidean distance between a data point in cluster $C_i \in \Lambda_k$ and a data point in cluster $C_j \in \Lambda_k$, given that $C_i \neq C_j$, and $diam(C_l)$ is the largest Euclidean distance between two data points in cluster $C_l \in \Lambda_k$. To find the optimal partition Λ_k^*, the Dunn Index should be maximized.

2.5.3 Spatially constrained hierarchical clustering

In some cases, it is needed to impose constraints on the possible outcomes of a hierarchical clustering process. When working with spatial data, it may be desired that the clusters are not only similar in data values, but also form spatially connected sets. For this, Chavent et al. (2018) developed a variation on the hierarchical clustering algorithm, called *spatially constrained hierarchical clustering*, which is summarized in this sub-section. The algorithm takes two dissimilarity matrices as input. The first one gives the dissimilarity values in the feature space, i.e. the dissimilarities of the data values of the observations, while the latter gives the dissimilarity values in the constraint space, i.e. the dissimilarities of the geographical locations of the observations.

Spatially constrained hierarchical clustering uses a Ward-like method to define which clusters will be merged at each step, but with a different definition of the information criterion of a cluster, as shown in Equation 2.x.

$$I(C) = (1-\alpha)\sum_{i=1}^{n}(c_i - \mu(C))^2 + \alpha\sum_{i=1}^{n}(c_i^* - \mu^*(C))^2$$

Where c_i are the data points in C, with normalized values taken from the feature space dissimilarity matrix, and $\mu(C)$ is the center of mass of C, computed with those values. c_i^* are the same data points, but with normalized values taken from the constraint space dissimilarity matrix, and $\mu^*(C)$ is the center of mass of C, computed with those values. Furthermore, n is the number of data points in C and α is the mixing parameter, where $0 \leq \alpha \leq 1$. Chavent et al. (2018) also present more general approaches, in which the calculated distances in Equation 2.x are not necessarily Euclidean, and where observations can be weighted, but these are not covered in this thesis.

With the information criterion of a single cluster calculated as in Equation 2.x, the information loss is of a merge between two clusters is calculated in the same way as in regular Ward hierarchical clustering (Equation 2.x). Then, at each merging step, two clusters are merged such that the loss of information is minimized.

The mixing parameter α plays a key role in spatially constrained hierarchical clustering. It sets the importance that is given to the spatial constraint. The higher α, the more the result of the clustering procedure is influenced by the spatial locations of the data points. When $\alpha = 0$, the data are clustered without any spatial constraint, while $\alpha = 1$ leads to a clustering only based on the spatial location of the data points. Therefore, it is important to determine a suitable value for α. Chavent et al. (2018) propose the following approach. First, *alpha* is set to zero, and a hierarchical clustering without spatial constraint is performed. The resulting sequence of partitions is rendered as a dendrogram, and the optimal number of clusters k^* is defined visually. Then, several spatially constrained hierarchical clusterings are performed, each with a different value of α, and $k = k^*$. Since k is fixed, the outputs are always single partitions. For each cluster in such a partition the information criterion regarding the feature data (i.e. the first part of Equation 2.x) and the information criterion regarding the constraint data (i.e. the second part of Equation 2.x), are calculated seperately, and also summed seperately over all clusters in the partition. These two summed values are then plotted, with α on the x-axis. In the end, this lead to a plot that shows the loss of information in the feature space and the gain of information in the constraint space, as α gets larger. With such a plot, α can be chosen such that the trade-off between the loss and gain of information in the two spaces is considered acceptable.

References

Box, George E. P., and Gwilym M. Jenkins. 1970. *Time Series Analysis: Forecasting and Control*. Book. Holden-Day San Francisco.

Brockwell, Peter J., and Richard A. Davis. 2002. *An Introduction to Time Series and Forecasting*. Vol. 39. doi:10.1007/978-1-4757-2526-1.

Cassisi, Carmelo, Placido Montalto, Marco Aliotta, Andrea Cannata, and Alfredo Pulvirenti. 2012. "Similarity Measures and Dimensionality Reduction Techniques for Time Series Data Mining." In. doi:10.5772/49941.

Chavent, Marie, Vanessa Kuentz-Simonet, Amaury Labenne, and Jérôme Saracco. 2018. "ClustGeo: an R package for hierarchical clustering with spatial constraints." *Computational Statistics* 33 (4): 1799–1822. doi:10.1007/s00180-018-0791-1.

Cleveland, Robert B., William S. Cleveland, Jean E. McRae, and Irma Terpenning. 1990. "STL: A Seasonal-Trend Decomposition Procedure Based on Loess." *Journal of Official Statistics* 6 (1): 3–73.

Cleveland, William S., and Susan J. Devlin. 1988. "Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting." *Journal of the American Statistical Association* 83 (403). Taylor & Francis: 596–610. doi:10.1080/01621459.1988.10478639.

Dunn, J. C. 1974. "Well-Separated Clusters and Optimal Fuzzy Partitions." *Journal of Cybernetics* 4 (1). Taylor & Francis: 95–104. doi:10.1080/01969727408546059.

Gan, Guojun, Chaoqun Ma, and Jianhong Wu. 2007. *Data Clustering: Theory, Algorithms, and Applications*. ASA-Siam Series on Statistics and Applied Probability. Society for Industrial; Applied Mathematics.

Hyndman, Rob J., and George Athanasopoulos. 2018. *Forecasting: principles and practice*. OTexts. https://otexts.org/fpp2/.

Hyndman, Rob J., and Yeasmin Khandakar. 2008. "Automatic time series forecasting: the forecast package for R." *Journal of Statistical Software* 26 (3): 1–22. http://www.jstatsoft.org/article/view/v027i03.

Keogh, Eamonn, and Jessica Lin. 2005. "Clustering of time-series subsequences is meaningless: Implications for previous and future research." *Knowledge and Information Systems* 8: 154–77. doi:10.1007/s10115-004-0172-7.

Kwiatkowski, Denis, Peter C. B. Phillips, Peter Schmidt, and Yongcheol Shin. 1992. "Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?" *Journal of Econometrics* 54 (1): 159–78. doi:https://doi.org/10.1016/0304-4076(92)90104-Y.

Ljung, Greta M., and George E. P. Box. 1978. "On a measure of lack of fit in time series models." *Biometrika* 65 (2): 297–303. doi:10.1093/biomet/65.2.297.

Nau, Robert. 2018. "Stationarity and differencing." https://people.duke.edu/{~}rnau/411diff.htm.

Shumway, Robert H., and David S. Stoffer. 2011. *Time Series Analysis and Its Applications With R Examples*. Vol. 9. doi:10.1007/978-1-4419-7865-3.

Wang, Xiaozhe Catherine, Kate Amanda Smith, and Rob J. Hyndman. 2006. "Characteristic-based clustering for time series data." *Data Mining and Knowledge Discovery* 13 (3). Springer: 335–64. doi:10.1007/s10618-005-0039-x.

Ward Jr., Joe H. 1963. "Hierarchical Grouping to Optimize an Objective Function." *Journal of the American Statistical Association* 58 (301). Taylor & Francis: 236–44. doi:10.1080/01621459.1963.10500845.

Woodward, Wayne A., Henry L. Gray, and Alan C. Elliott. 2017. *Applied Time Series Analysis with R*. CRC Press.