MASTER

Data based improvement of inventory forecasting

Heuvelman, Dionne M.

Award date:
2020

Eindhoven University of Technology

Graduation project at Bottomline

Industrial and Applied Mathematics

Data based improvement of inventory forecasting

Author:
Dionne Heuvelman
d.m.heuvelman@student.tue.nl
0959460

Supervisor TU/e:
Edwin van den Heuvel
e.r.v.d.heuvel@tue.nl

Supervisor TU/e:
Marta Regis
m.regis@tue.nl

Supervisor Bottomline:
Tijmen van Dien
tijmen.vandien@bottomline.eu

December 16, 2020

Abstract

The focus of this research is if the inventory forecasts of Bottomline can be improved, and if possible how this can be achieved. To investigate this, literature research is done and multiple methods like Holt-Winters and ARMA are implemented. To have non-negative forecasts, a cut-off below zero and a Box-Cox transformation are investigated. To improve the forecasts, a bootstrapping method is applied. It is found that all methods calculate the best forecasts when 30 weeks of historic observations are used. The forecasts of the ARMA method with cut-off below zero have the smallest Mean Squared Error, and therefore the advice to Bottomline is to use the ARMA method with cut-off with 30 weeks of historic data.

Contents

1 Introduction

Automated processes to support or replace humans are growing rapidly worldwide. One of the companies working hard on this topic is Bottomline, which makes software to realize automated real time inventory routing. The company works in the field of inventory routing and transport for products like diesel and petrol. The customers of Bottomline are petrol stations and it is the responsibility of Bottomline that the product storages never run out of product. At the company Bottomline, the inventory of the petrol stations has to be predicted to make sure that trucks deliver products at the stations before the storages become empty (otherwise there will be a penalty). To help to realize this, Bottomline has its own software to help the planners (people who schedule the trucks based on predicted inventory levels) and truck drivers with the administration. There is also a development team that is trying to make software that can create plans automatedly, but this is not ready to use yet.
A very important aspect of the inventory routing, is the inventory forecasting. This has to be done for every petrol station and for every type of product present at the station. When the inventory forecasts are reliable and there are only small errors, this really helps the planners in their job, and this reduces penalties for the company.
At the moment, the calculations of the inventory forecasts of the company are relatively simple, and they sometimes underestimate the inventory (at purpose to prevent penalties). This means that the prediction of the inventory is lower than the real inventory, and it is not always possible to deliver the planned amount of product. This causes that there will be product left in the truck after all deliveries, and the driver has to call a planner where to go next to deliver the remaining product. It would be beneficial if the inventory forecasts could be more accurate and left-over product in the truck less often. Therefore it will be investigated if it is possible to calculate more accurate inventory forecasts, and how this can be done.
There are some mathematical challenges in this project, like the heterogeneity of the storages. There are a lot of storages and they do not all have the same properties. This can make it challenging to find one best forecasting method. Another mathematical challenge is that the data can contain zero values, which occur when a storage is closed and there are no sales that day. Some forecasting methods can not deal with zero values so this will be a point of interest. To be able to determine if the forecasts can be improved, it has to be determined on what aspect a forecasts will be scored.
To investigate how to improve the inventory forecasts, first the current method and the data have to be analysed in detail. Next a literature research will be done to investigate which methods can be used, what advantages and disadvantages they have, and if possible, how to overcome those disadvantages. The found methods will be applied to the data and it will be evaluated which method can be used best to predict the data. Also, the new methods will be compared to the currently used method to see if a new method can be used to calculate more accurate forecasts, where the objective is to make a move towards solving the problem of the company, both in theory and in practice.

1.1 Mathematical formulation and data description

To improve the forecasts, data of the company can be used. There is a lot of data available at Bottomline, and the data that can be used for this project is described in this section. The information will be described for one product at one station, and the methods will be evaluated univariately.
The data exists of information about a lot of storages that are divided over different sites. For every storage some properties are known like the site where the storage is located, what type of product it contains, how many weeks to use for calculating the forecasts and what the weekly and daily sales patterns are.
The weekly sales pattern describes the percentage of the sales of the week that is sold per every day of the week, and is denoted by P_i where i is the day of the week. A simply example of a weekly pattern is $[20, 20, 20, 20, 20, 0, 0]$ which would mean that on Monday 20% of the weekly sales is sold, so $P_1 = 20$, on Tuesday 20% is sold, and also on Wednesday, Thursday and Friday, and during the weekend there are no sales.
The daily sales pattern gives a percentage of the daily sales that is sold during a period and is denoted by p_i, where i is the number of the period of a day. For example when a day is divided in 4 equal periods, an example of a daily pattern is $[0, 50, 50, 0]$. This means that $p_1 = 0$, so in the first 6 hours of the day there

are no sales. In the next 6 hours, so from $6.00h$ to $12.00h$ half of the daily sales take place, and so on.
Next to the described properties, there are some data sets available that contain information of the following properties over time.

- A data set containing the inventory measurements V_t at time t, for the complete history of the storage.
- A data set containing the delivery quantities U_t at time t, for the complete history of the storage.
- A data set containing the knowledge information K_i for day i, for the complete history of the storage and the knowledge for the future that is already available. Knowledge information can be one of two types, an absolute value or a multiplication factor. This will be further explained in the Current Method in Section 2

Based on the data over time of these properties, the sales per day are calculated. The data described is no real time data because sometimes the driver of a truck performs a measurement and has to submit this by hand, so there is a delay in the data over time.

To calculate the sales per day, the data is analysed per 2 inventory measurements, and this is repeated for the whole time period for which the sales per day need to be calculated.
Let two inventory measurements be at time t_1 and t_2. For the time between two inventory measurements, it is calculated what the sales per day were. To calculate the sales per day, the date is important, but it is also important which day of the week is was, for example Monday, because this is needed to use the weekly pattern. The calculation of the sales per day is done in the following way:
First the total sales during the period between time t_1 and t_2 is calculated. This is done by calculating the difference between the inventories, and if there were one or more deliveries in the meantime, this amount has to be added to calculate the total sales.

$$w_{t_1,t_2} = V_{t_1} - V_{t_2} + \sum_t U_t \cdot \mathbb{1}_{t\in(t_1,t_2)}$$

where w_{t_1,t_2} is the total sales between time t_1 and time t_2, U_t is the delivery quantity at time t and $\mathbb{1}$ is the indicator function.
Once the total sales in the period between t_1 and t_2 is known, it can be calculated what part of the sales was on which day, using the weekly and daily sales pattern. Let the percentage of the sales that was sold on day i be M_i. That way it can be calculated what the daily sales were.

$$W_i(t_1, t_2) = M_i \cdot w_{t_1,t_2}$$

Where $W_i(t_1, t_2)$ are the sales on day i based on the inventory measurements at time t_1 and t_2.
To calculate the total sales on day i denoted by W_i, for example when t_2 was at the middle of day i, the procedure has to be repeated, and $W_i = W_i(t_1, t_2) + W_i(t_2, t_3)$.
This way all the daily sales for an historic period can be calculated, and this can be used to predict daily sales for the future. This can then be used in combination with the last inventory measurement to estimate the point in time that the inventory will run out, and thus before which point in time the next delivery has to be scheduled.

Note that every time, two consecutive inventory measurements are used, and the weekly pattern is used when there are multiple days between the two measurements. This way it is not possible to have missing values in the daily sales observations.

The goal of this report is to investigate if the sales forecasts can be improved, an how to do this if possible. Therefore multiple forecasting methods will be investigated and implemented. The accuracy of the forecasts will be evaluated using the Mean Squared Error, which is defined as follows.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(W_i - \hat{W}_i)^2$$

where W_i is the observed sales for day i and $\hat{W}_i$ is the predicted sales for day i. This way it can be determined which method produces the most accurate forecasts.

2 Current forecasting model at Bottomline

In the current forecasting method, the sales per day are used to forecast the sales for a day in the future. For every storage a value is set that determines the number of weeks in the past that is used to forecast the sales, and this number is denoted by y. The forecasting of the sales is done in several steps:

1. **Remove outliers**
   When a value deviates too much from the mean, where the variability is considered, it is assumed to be an outlier. Whether a value deviates too much is determined using two rules. For both rules, the value of the sales is compared to the mean sales for that day of the week.
   The first rule is that
   $$W_i \notin [\mu - b\sigma, \mu + b\sigma]$$
   where μ is the mean sales per day of a storage, and σ is the standard deviation of the sales per day of a storage, and b is a value determined for all the storages in a country.
   The second rule is that
   $$W_i \notin [\mu - c\mu, \mu + c\mu]$$
   where μ is the mean sales of a storage and c is a factor determined for all the storages in a country.
   It can be set if a value is an outlier if both rules are satisfied, or if a value is an outlier if at least one of the rules is satisfied.
   From the sales values of the past y weeks, the outlier values are removed for all the following steps, and not replaced by some other value. The missingness of some data points is not problem for this forecasting method.
   The only exception for removing outlier values is when all values of one day (for example all Mondays) are marked as outliers, it would not be possible to forecast sales on Monday, so in that case the values of that day will not be removed.

2. **Calculate mean sales per week**
   The mean sales per week is calculated using the sales per day.
   The mean sales value of the week that ends at day i is:
   $$\overline{W}_i = \frac{W_i + W_{i-1} + W_{i-2} + W_{i-3} + W_{i-4} + W_{i-5} + W_{i-6}}{7}$$
   where W_{i-1} is the sales value of one day before day i.

3. **Calculate weekly fluctuations**
   Using the mean sales per week, the average sales for all the y weeks together is calculated.
   The goal of Bottomline is to determine the weekly pattern, and in order to do this the fluctuations in the mean sales per week are calculated and corrected for.
   The fluctuation factor per week is calculated, where a high fluctuation factor means that there were relatively high sales that week, compared to the average sales. This is useful because the weekly pattern is important to be able to have a forecast that is close to reality. The fluctuation factor f_j of the week that ends at day j is calculated by dividing the sales of the week by the average sales of all weeks.
   $$f_j = \frac{\overline{W}_j}{\frac{1}{y}\sum_{k=1}^{y} \overline{W}_k}$$
   For example, if $y = 2$, and the average sales of 1 week ago is 2, and the average sales of 2 weeks ago is 3, then the total average sales of all weeks is 2.5. Then the seasonality factor for 1 week ago is $2/2.5 = 0.8$. The seasonality factor for 2 weeks ago is $3/2.5 = 1.2$.

4. **Correct daily sales for fluctuations**
   To correct for the fluctuations, the daily sales are divided by the fluctuation factor of the corresponding week.
   $$\hat{W}_i = \frac{W_i}{f_j}$$
   where $\hat{W}_i$ is the corrected sales for day i, and day i is part of week j. The corrected sales values will be used in the rest of the forecasting procedure.

5. **Calculate mean sales per day of the week based on corrected values of previous weeks**
   To calculate the predicted sales for a day, the mean of the corrected values of the last y weeks is calculated.
   For example, when $y = 2$ and the last two corrected Wednesday sales were 0.6 and 0.7, the prediction for the sales for next Wednesday will be set to 0.65. So the forecasting value for day i and $y = 2$ is: $\tilde{W}_i = ((\hat{W}_{i-7} + \hat{W}_{i-14}))/2$, where $\hat{W}_{i-7}$ is the corrected sales value 7 days ago.

6. **Apply market knowledge if present**
   For some days it is given by the owner of the storage that the expected sales deviate from the normal sales for that day, and this information is called knowledge. This can be given as an absolute number or as a percentage. When an absolute number is given, this number will be used to replace the predicted sales to achieve the final forecast value. When a percentage is given, the predicted sales are multiplied by the factor (percentage divided by 100) to obtain the final forecast value. When no market knowledge is given for a day, the predicted sales are the final forecast value.

Next the inventory forecast is used to predict run-out and run-dry moments of a storage to determine when to schedule the next delivery. Run-out means that the storage has reached its safety stock level, the minimum amount the storage owner wants to have in his storage, and the run-out moment means that the storage is completely empty.

3 Methods

In Section 1.1 is described how the sales per day are calculated. In Chapter 2 is seen how the sales per day are currently used to make forecasts at Bottomline. In this Chapter will be focused on other methods to calculate forecasts, based on the sales per day from Section 1.1. The sales are data points which are measured over time, with equal time intervals between measurements. Therefore the sales are a time series, and due to the use of the weekly pattern, there are no missing values in the time series. To be able to predict values in the future, two widely used forecasting methods, exponential smoothing and autoregressive moving average methods, are investigated in this chapter [1] [2] [3].

3.1 Remove zero values of holidays

The focus of this project will be the sales per day which are calculated from the data of Bottomline. Note that the sales are always non-negative. Zero values mean that there are no sales on that day, so the storage was closed, and this can happen for two reasons. The first reason zeros are present is on days when a storage is closed systematically, for example when a storage is always closed on Sunday, and the second reason is when the storage is closed for some other reason, for example on a holiday. The zeros that occur on regular closing days like Sundays are useful to predict the future sales for that day. However, the zeros on holidays can disturb the forecasts because they are not regular and do not have to be predicted for the future. If there is another holiday in the future, this will be covered with the knowledge for that day. Therefore it is best to remove the zeros that are due to holidays. A way to determine that they belong to a holiday is to remove zeros that are set by knowledge.
Once the zero values set by knowledge are removed there are two options to deal with the missing values. If the forecasting method can deal with missing values, it is best to keep it that way, because any adjustment leads to extra uncertainty and possible errors.

However, when the forecasting method can not deal with missing values it is necessary to replace the missing value. When a missing value is not replaced, not every week has an equal amount of observations and this is a problem when dealing with seasonality because one has to set a fixed length of the seasonality period. Therefore a missing value has to be replaced and this can be done by imputing the mean of the other observations of the same day from the data set. This mean value will be closer to the 'normal' sales for that day than the zero value due to the holiday, and therefore this will give a better forecast.
The mean value of a day that will be imputed is calculated as follows.

$$\underline{W}_i = \frac{1}{y-1}(\sum_{j=1}^{k-1} W_{7j+l} + \sum_{j=k+1}^{y} W_{7j+l})$$

Where the missing value is at day i (which is in week k, and is day l of the week, where Monday is day 1 of the week), and y weeks of data are used to calculate the forecasts. The value that will be imputed for the missing sales value at day i is $\underline{W}_i$.

The zero values on holidays are not necessary to predict zero values for future holidays, because for future holidays new knowledge will be given. Therefore removing zero values from historic observations that were set by knowledge will not cause problems, and it can reduce the forecasting error.

3.2 Time series

A time series is a set of observations, where each observation is recorded at a specific point in time. A time series is notated as $\{X_t\}$ where X_t is the t^{th} observation and $t = 0, 1, ...$ is the index for the time. A time series is a realization of a stochastic process $\{Y_t\}$. A stochastic process has some important properties, which are the mean,

$$\mu_t = \mathbb{E}[Y_t]$$

the variance,

$$\sigma_t^2 = Var(Y_t)$$

and the autocovariance

$$\gamma(t_1, t_2) = \mathbb{E}[(Y_{t_1} - \mu_{t_1})(Y_{t_2} - \mu_{t_2})]$$

With forecasting time series data, the goal is to estimate how the sequence of observations will continue in the future. A time series can be decomposed in a level, a trend, seasonal terms and random terms. The level is the average value of the series. The trend gives the increase or decrease of the values of the series, which is the long-term variation. The seasonality describes a repeating short-term cycle in the series. The random terms are the random variations in the series, and this is the only non-systematic term.
There are two types of seasonality, additive and multiplicative. With additive seasonality, it is assumed the observations can be decomposed as

$$X_t = L_t + T_t + S_t + \epsilon_t$$

with L_t is deseasonalized mean level, T_t is the trend effect, and S_t is seasonal effect, and those are all functions of time. The random effect is represented by ϵ_t. Additive seasonality means that independent of the mean level, the absolute difference in the observed values due to the seasonal effect is the same.
With multiplicative seasonality, the observations are constructed as

$$X_t = (L_t + T_t) \cdot S_t \cdot \epsilon_t$$

With a multiplicative seasonal effect, the absolute difference in the observed values due to the seasonal effect increases as the mean level increases.
The difference between an additive and a multiplicative model is visualized in the figure below.

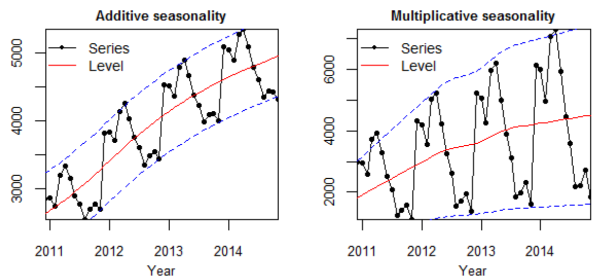

Figure 1: Additive model versus Multiplicative model

3.2.1 Stationary process

When a process does not have trend and seasonality, it is called a stationary processes. Stationary processes are processes whose properties do not depend on the time at which the process is observed. This means that stationary processes have no predictable patterns, and have constant variance. Formally this can be defined as a time series $\{X_t\}$ for which shifting the time origin by τ has no effect on the joint cumulative probability distribution. So a time series is stationary when

$$F_X(x_{t_1}, x_{t_2}, ..., x_{t_n}) = F_X(x_{t_1+\tau}, x_{t_2+\tau}, ..., x_{t_n+\tau})$$

for all $t_1, t_2, ..., t_n, \tau \in \mathbb{R}$ and $n \in \mathbb{N}$.
Autocovariance γ and autocorrelation ρ properties for stationary processes are defined as follows.

$$\gamma(t_1, t_2) = \gamma(\tau) = \mathbb{E}[(X_t - \mu)(X_{t+\tau} - \mu)] = Cov(X_t, X_{t+\tau}) \tag{1}$$

$$\rho(\tau) = \gamma(\tau)/\gamma(0) \tag{2}$$

Where μ is the mean value of the process.
When a series contains trend, there is a long-term change in the mean level. Finite differencing is a method to remove the trend, and obtain a stationary time series. The aim is to remove the trend to analyse local fluctuations. Finite differencing is defined as

$$W_t = \nabla X_t = X_t - X_{t-1}$$

where the result is time series W_t which is stationary. When the trend is non-linear, higher order differencing might be needed, for example second order differencing, which is defined as follows.

$$\nabla^2 X_t = \nabla X_t - \nabla X_{t-1} = X_t - 2X_{t-1} + X_{t-2}$$

When a series contains seasonality, seasonal decomposition can be applied to calculate the deseasonalized mean level m_t of the series. For example for weekly data, so with seasonality period $s = 7$

$$m_t = \frac{x_{t-3} + x_{t-2} + x_{t-1} + x_t + x_{t+1} + x_{t+2} + x_{t+3}}{7}$$

When the deseasonalized mean level is calculated, this can be used to estimate the seasonal effect S_t of the series. For additive seasonality, the seasonal effect is $S_t = X_t - m_t$ and for multiplicative seasonality, the seasonal effect is $S_t = X_t/m_t$.
To remove the seasonality and obtain a stationary series, seasonal differencing can be used. For seasonal period s, seasonal differencing is defined as

$$W_t = \nabla_s X_t = X_t - X_{t-s}$$

where the result is time series W_t which is stationary. For a non-linear seasonality pattern, for example multiplicative seasonality, higher order differencing might be needed.
When a series contains both trend and seasonality, finite differencing and seasonal differencing can both be applied to obtain a stationary time series.

3.3 Exploratory Data Analysis

Before modelling a time series, more information has to be obtained about properties like if there are trend and seasonality. To analyse this, some visualizations might help, like the time sequence plot, autocorrelation function plot and partial autocorrelation plot. This first stage of gathering information is called the exploratory data analysis.
First the time sequence plot can be analysed, focusing on the main properties: trend, cyclic variation and irregular variation. Next it can be investigated if there are sudden changes, outliers and missing values.
The figure below is an example of a time sequence plot of daily data, where the daily measurements are visualized over time. There is a sales observation every day, and on the x-axis the weeks are visualized. When executing exploratory data analysis, it can be noted that the mean value is not constant over time, and in that case there might be a trend. The first 12 weeks there is a linear upward trend, but after week 12 there is an exponential decay. This data starts at January 1st of 2020, so the exponential decay is due to the corona crisis. Next to the trend, there seems to be a weekly pattern in the data, so there is seasonality present. However the seasonality pattern might be not fixed.

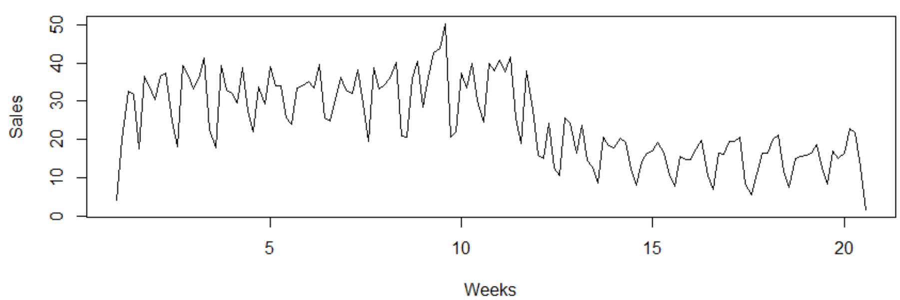

Figure 2: Example of a time sequence plot

It is possible that multiple seasonality periods are present in one time series, for example a short-term seasonality that describes a weekly pattern and a long-term seasonality that describes a monthly or yearly pattern. A yearly seasonality pattern can easily be confused with a trend pattern. Those are things to keep in mind when performing an exploratory data analysis.

3.3.1 Time series decomposition

Once the time sequence plot is analysed and aspects of trend and seasonality may be found, more information can be gathered by performing a time series decomposition. In a time series decomposition, the components of a time series like trend and seasonality can also be decomposed and visualized separately. When there is uncertainty about the seasonality, time series decomposition with and without seasonality (or different seasonality periods) can be applied and it can be evaluated in which case the remainder terms are the smallest, because this is likely to be the best model.
The classical time series decomposition is performed as follows.
First the trend T is detected using a moving average window of the size of the seasonality. Next the detrended series is calculated. For additive time series, this is done by subtracting the trend from the original time series.

$$D_t = X_t - T_t \tag{3}$$

Where D_t is the detrended observation value at time t, X_t is the original observation value and T_t is the calculated trend component at time t.
For multiplicative time series, the detrended observations are calculated in the following way.

$$D_t = X_t / T_t \tag{4}$$

The seasonality component of the time series is calculated by averaging the detrended observations by seasonality period. This means that in this case, the average of all Monday observations is calculated, and this is the seasonality factor for Mondays. For example when there are n weeks of observations and t is in the last week, the following is obtained.

$$S_t = \frac{\sum_{i=1}^{n} D_{t-7i}}{n} \tag{5}$$

This is done for all days, and then the seasonality component is obtained.
Finally, the random components are calculated by removing the trend and the seasonality components from the original series. For an additive time series (in both trend and seasonality), the random component is calculated as follows.

$$R_t = X_t - T_t - S_t$$

Where R_t is the random component at time t and S_t is the calculated seasonality component at time t. For a time series with multiplicative seasonality this is done in the following way.

$$R_t = (X_t - T_t)/S_t \tag{6}$$

This way all the time series components are extracted from the series and can be shown separately. This procedure can be automatically performed in R using the *decompose* function from the package *EnvStats* [4].

Another, more modern time series decomposition method is the STL method. The STL method for time series decomposition is an abbreviation for Seasonal and Trend decomposition using Loess, and Loess is a nonlinear estimation method [5]. STL is a robust method for additive time series decomposition and allows the seasonality component to change over time. For STL, a trend-cycle window and a seasonal window have to be chosen. Smaller values allow for more rapid changes in the trend and seasonality components.
In the STL method, the following steps can be separated. First the series is detrended as in Formula 3. Next for each seasonal component (for example for every day in data with a weekly seasonality pattern) a subseries is constructed and averaged, just like in Formula 5. Then the subseries are put together again and the total seasonal component is found. Then the calculated seasonal component is used to deseasonalize the original time series. Lastly the deseasonalized series is smoothed to find the trend component. This method is robust to outliers which means that outlier observations will not affect the estimates of the trend and seasonal components, but will affect the residual component.

The main difference between the two time series decomposition methods is that the STL method allows for changes in the seasonality component over time. A disadvantage of the STL method is that it only deals with additive data, and there is no variant for multiplicative data. However in some cases of multiplicative data it is possible to apply an logarithmic transformation to obtain an additive time series and STL can be applied.

3.4 Exponential smoothing models

Exponential smoothing methods can be used to calculate a one-step ahead prediction. This is done by calculating a weighted sum of past observations.

$$\hat{x}_t(1) = c_0 x_t + c_1 x_{t-1} + c_2 x_{t-2} + ... \tag{7}$$

Where $\hat{x}_t(1)$ is the one-step ahead prediction from time t, c_i's are the weights and x_t is the observation at time t.
For exponential smoothing models, the weights decrease exponentially as the observations are further in the past. Therefore geometric weights are used.

$$c_i = \alpha(1-\alpha)^i \tag{8}$$

Where α is the smoothing parameter, a value that is chosen between zero and one, which controls the rate at which the weights decrease. The exponentially decreasing weights ensure that recent information has a bigger contribution to the predictions compared to past information. This combination of Formulas 7 and 8 leads to the following prediction formula for exponential smoothing methods.

$$\begin{aligned}\hat{x}_t(1) &= \alpha x_t + \alpha(1-\alpha)x_{t-1} + \alpha(1-\alpha)^2 x_{t-2} + ... \\ &= \alpha x_t + (1-\alpha)(\alpha x_{t-1} + \alpha(1-\alpha)x_{t-2} + ...) \\ &= \alpha x_t + (1-\alpha)\hat{x}_{t-1}(1)\end{aligned} \tag{9}$$

When the smoothing parameter α is chosen close to 0, more weight is given to observations further in the past, which is called slow learning. When α is chosen close to 1, more weight is given to the more recent observations, which is called fast learning.
When applying an exponential smoothing method, the value of $\hat{x}_t(1)$ is calculated in multiple steps by updating the individual components first and then combining this to calculate $\hat{x}_t(1)$. In the following subsections, multiple exponential smoothing methods are explained with the corresponding formulas, which are all based on Formula 9.
When a multiple-step ahead prediction is wanted, the one-step ahead prediction can be applied iteratively, where the future observations are replaced by their forecasts.

3.4.1 Simple exponential smoothing

Simple exponential smoothing is a forecasting method for stationary time series, so series that have no clear trend or seasonal behaviour. Simple exponential smoothing forecasting gives a level estimate for the time series, referred as $\hat{L}$, which is the expected value of the next observation, around which some randomness takes place. Simple exponential smoothing allows for updates of level estimates, which is done in the following way. This is a rewriting of Formula 9, and will be extended in the following subsections.

$$\begin{aligned}&\hat{x}_{t-1}(1) = \hat{L}_{t-1} \\ &\hat{L}_t = \alpha x_t + (1-\alpha)\hat{L}_{t-1}\end{aligned} \tag{10}$$

Where $\hat{x}_{t-1}(1)$ is the one-step ahead prediction at time $t-1$, $\hat{L}_t$ is the level estimate for time t, and x_t is the observation at time t. Every timestep, the forecast for the next step is calculated, and the level estimate is updated, which will be used in the next timestep. An adequate initial value for the level estimate can be chosen as $\hat{L}_1 = x_1$.

3.4.2 Holt's exponential smoothing

Holt's exponential smoothing is a forecasting method for time series that have trend, but no seasonality. Holt's exponential smoothing principle is as follows.

$$\begin{aligned}&\hat{x}^H_{t-1}(1) = \hat{L}^H_{t-1} + \hat{T}^H_{t-1} \\ &\hat{L}^H_t = \alpha x_t + (1-\alpha)(\hat{L}^H_{t-1} + \hat{T}^H_{t-1}) \\ &\hat{T}^H_t = \beta(\hat{L}^H_t - \hat{L}^H_{t-1}) + (1-\beta)\hat{T}^H_{t-1}\end{aligned} \tag{11}$$

Where $\hat{x}^H_{t-1}(1)$ is the one-step ahead Holt's exponential smoothing forecast for time $t-1$, $\hat{L}^H_t$ is the Holt's level estimate for time t, $\hat{T}^H_t$ is the Holt's trend estimate for time t, and x_t is the observation at time t. The smoothing parameter for the level is α, and the smoothing parameter for the trend is β. Holt's exponential smoothing allows for updates of level and trend estimates. Adequate initial values for the level and trend estimates can be chosen as $\hat{L}_1 = x_1$ and $\hat{T}_1 = x_2 - x_1$.

3.4.3 Simple seasonal exponential smoothing

Simple seasonal exponential smoothing is a forecasting method for time series that have no trend, but do have additive seasonality. Simple seasonal exponential smoothing principle is as follows.

$$\begin{aligned} \hat{x}^S_{t-1}(1) &= \hat{L}^S_{t-1} + \hat{S}^S_{t-s} \\ \hat{L}^S_t &= \alpha(x_t - \hat{S}^S_{t-s}) + (1-\alpha)\hat{L}^S_{t-1} \\ \hat{S}^S_t &= \gamma(x_t - \hat{L}^S_t) + (1-\gamma)\hat{S}^S_{t-s} \end{aligned} \tag{12}$$

Where $\hat{x}^S_{t-1}(1)$ is the one-step ahead Simple seasonal exponential smoothing forecast for time $t-1$, $\hat{L}^S_t$ is the level estimate for time t, $\hat{S}^S_t$ is the seasonality estimate for time t, and x_t is the observation at time t. The smoothing parameter for the level is α, and the smoothing parameter for the seasonality is γ. Simple seasonal exponential smoothing allows for updates of level and seasonality estimates. Adequate initial values for the level and trend estimates for the additive model can be chosen as

$$\hat{L}_1 = \frac{x_1 + x_2 + ... + x_s}{s}$$

$$\begin{bmatrix} \hat{S}_1 \\ \hat{S}_2 \\ \vdots \\ \hat{S}_s \end{bmatrix} = \begin{bmatrix} x_1 - \hat{L}_1 \\ x_2 - \hat{L}_1 \\ \vdots \\ x_s - \hat{L}_1 \end{bmatrix}$$

where s is the length of the seasonality period.
Adequate initial values for the level and trend estimates for the multiplicative model can be chosen as

$$\hat{L}_1 = \frac{x_1 + x_2 + ... + x_s}{s}$$

$$\begin{bmatrix} \hat{S}_1 \\ \hat{S}_2 \\ \vdots \\ \hat{S}_s \end{bmatrix} = \begin{bmatrix} x_1/\hat{L}_1 \\ x_2/\hat{L}_1 \\ \vdots \\ x_s/\hat{L}_1 \end{bmatrix}$$

where s is the length of the seasonality period.

3.4.4 Holt-Winters exponential smoothing

Holt-Winters exponential smoothing is a forecasting method for time series that contain both trend and seasonality. For series with additive seasonality, the forecasting principle is as follows.

$$\begin{aligned} \hat{x}^{HW}_{t-1}(1) &= \hat{L}^{HW}_{t-1} + \hat{T}^{HW}_{t-1} + \hat{S}^{HW}_{t-s} \\ \hat{L}^{HW}_t &= \alpha(x_t - \hat{S}^{HW}_{t-s}) + (1-\alpha)(\hat{L}^{HW}_{t-1} + \hat{T}^{HW}_{t-1}) \\ \hat{T}^{HW}_t &= \beta(\hat{L}^{HW}_t - \hat{L}^{HW}_{t-1}) + (1-\beta)\hat{T}^{HW}_{t-1} \\ \hat{S}^{HW}_t &= \gamma(x_t - \hat{L}^{HW}_t) + (1-\gamma)\hat{S}^{HW}_{t-s} \end{aligned} \tag{13}$$

For series with multiplicative seasonality, the forecasting principle is as follows.

$$\begin{aligned} \hat{x}^{HW}_{t-1}(1) &= (\hat{L}^{HW}_{t-1} + \hat{T}^{HW}_{t-1})\hat{S}^{HW}_{t-s} \\ \hat{L}^{HW}_t &= \alpha(x_t/\hat{S}^{HW}_{t-s}) + (1-\alpha)(\hat{L}^{HW}_{t-1} + \hat{T}^{HW}_{t-1}) \\ \hat{T}^{HW}_t &= \beta(\hat{L}^{HW}_t - \hat{L}^{HW}_{t-1}) + (1-\beta)\hat{T}^{HW}_{t-1} \\ \hat{S}^{HW}_t &= \gamma(x_t/\hat{L}^{HW}_t) + (1-\gamma)\hat{S}^{HW}_{t-s} \end{aligned} \tag{14}$$

Where $\hat{x}_{t-1}^{HW}(1)$ is the one-step ahead Holt-Winters exponential smoothing forecast for time $t-1$, $\hat{L}_t^{HW}$ is the level estimate for time t, $\hat{T}_t^{HW}$ is the trend estimate for time t, $\hat{S}_t^{HW}$ is the seasonality estimate for time t, and x_t is the observation at time t. The smoothing parameter for the level is α, the smoothing parameter for the trend is β, and the smoothing parameter for the seasonality is γ. Holt-Winters exponential smoothing allows for updates of level, trend and seasonality estimates. Adequate initial values for the level and trend estimates for the additive model can be chosen as

$$\hat{L}_1 = \frac{x_1 + x_2 + ... + x_s}{s} \tag{15}$$

$$\hat{T}_1 = \frac{(x_s + x_{s+1} + ... + x_{s+s}) - (x_1 + x_2 + ... + x_s)}{s^2} \tag{16}$$

$$\begin{bmatrix} \hat{S}_1 \\ \hat{S}_2 \\ \vdots \\ \hat{S}_s \end{bmatrix} = \begin{bmatrix} x_1 - \hat{L}_1 \\ x_2 - \hat{L}_1 \\ \vdots \\ x_s - \hat{L}_1 \end{bmatrix} \tag{17}$$

where s is the length of the seasonality period.
Adequate initial values for the level and trend estimates for the multiplicative model can be chosen as

$$\hat{L}_1 = \frac{x_1 + x_2 + ... + x_s}{s} \tag{18}$$

$$\hat{T}_1 = \frac{(x_s + x_{s+1} + ... + x_{s+s}) - (x_1 + x_2 + ... + x_s)}{s^2} \tag{19}$$

$$\begin{bmatrix} \hat{S}_1 \\ \hat{S}_2 \\ \vdots \\ \hat{S}_s \end{bmatrix} = \begin{bmatrix} x_1 / \hat{L}_1 \\ x_2 / \hat{L}_1 \\ \vdots \\ x_s / \hat{L}_1 \end{bmatrix} \tag{20}$$

where s is the length of the seasonality period.

3.4.5 Missing values

When using the exponential smoothing model in R to calculate forecasts, is it not possible to have time series that contain missing values. Therefore the missing values that arise when removing zero values from holidays (set by knowledge) will be replaced by the mean value of the observations of the same weekday in the data set, as described in Section 3.1.

3.5 ARMA models

Another very popular way to model time series, is using autoregressive moving average (ARMA) models. Where exponential smoothing models were based on a description of trend and seasonality in the data, ARMA models aim to describe the correlations in the data.
ARMA models can be used for the modelling of processes that are influenced by both levels and disturbances from the past. The autoregressive part predicts the average level through regression, and the moving average part predicts the noise. The ARMA models were introduced by Box and Jenkins, and therefore often referred as Box-Jenkins models. To understand the Box-Jenkins models, first the autoregressive model and moving average model are studied individually.

Two very useful visualisations that can be helpful for ARMA models are the autocorrelation function and the partial autocorrelation function. The autocorrelation is a visualization of the autocorrelation for different lags k, where the autocorrelation is a measure for the correlation between X_{t-k} and X_t. The partial autocorrelation function is a measure for the direct correlation between X_{t-k} and X_t, corrected for the indirect

correlation.
An example of an autocorrelation function (ACF) plot and partial autocorrelation function (PACF) plot, referred to the time sequence plot in Figure 2, can be seen in the figure below.

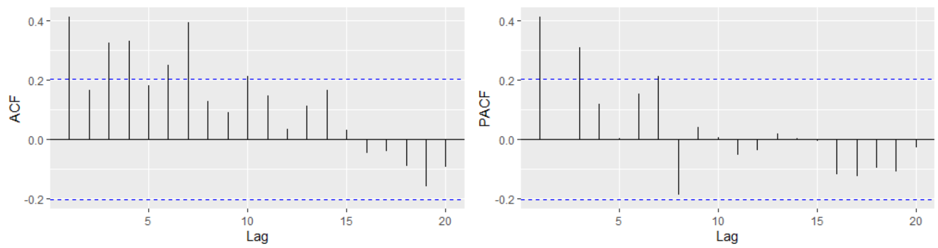

Figure 3: Example of an ACF plot and PACF plot

In both the ACF and PACF there is a large contribution for lag 7 for example, which indicates a weekly seasonality effect (when daily data is used). Further analysis of the ACF and PACF plots will be described later.

3.5.1 Autoregressive AR(p) process

The autoregressive process can be used to model a stationary time series, and is best applicable for time series which have longer dependencies between observations. This means that it is suitable to model a process that is mainly influenced by past values of the process. The autoregressive process with order p is denoted by

$$X_t = \alpha_1 X_{t-1} + ... + \alpha_p X_{t-p} + Z_t$$

Where X_t is the observation at time t, and Z_t is the i.i.d. white noise at time t. The values of a white noise process are normally distributed with mean 0 and variance σ_Z^2.
Let the backshift operator $\boldsymbol{B}$ be such that

$$\boldsymbol{B}X_t = X_{t-1} \text{ and } \boldsymbol{B}Z_t = Z_{t-1}$$

When the backshift operator is applied iteratively, the following is obtained.

$$\boldsymbol{B}^p X_t = X_{t-p} \text{ and } \boldsymbol{B}^q Z_t = Z_{t-q}$$

The autoregressive process can also be formulated as follows, using the backshift operator.

$$\phi_p(\boldsymbol{B})X_t = Z_t \tag{21}$$

With $\phi_p(\boldsymbol{B}) = 1 - \alpha_1 \boldsymbol{B} + ... + \alpha_p \boldsymbol{B}^p$.
For the autoregressive process, the Yule-Walker equation holds, which states the following.

$$\gamma(k) = \mathbb{E}[X_t X_{t+k}] = \mathbb{E}[X_t(\alpha X_{t+k-1} + Z_{t+k})]$$

This can be used to calculate the autocorrelation. The AR(1) process has the following properties

$$\mathbb{E}[X_t] = 0$$

$$Var(X_t) = \sigma_Z^2 \sum_{i-0}^{\infty} \alpha_i^{2i} = \frac{\sigma_Z^2}{1-\alpha^2}$$

$$\rho(k) = \alpha^{|k|} \text{ for } k = 0, 1, -1, 2, -2, ...$$

A special property of the autoregressive process is that the partial autocorrelations for $k > p$ are equal to zero. This property can be used when estimating the value of the order p, because in the partial autocorrelation function a sudden drop should be seen after lag p, which indicates the value of order p. Another rule of thumb is that in the autocorrelation function, exponentially decaying behaviour can be seen after lag p.
There condition that has to be satisfied to have a stationary process is $\phi_p(\boldsymbol{B}) \neq 0$. This means that the solutions of $\phi_p(\boldsymbol{B}) = 0$ have to be outside the complex unit circle.

3.5.2 Moving average MA(q) process

The moving average process is a way to model a stationary time series with short term dependencies between observations. The time series have to be stationary, so there has to be no trend or seasonality. The moving average process is denoted by

$$X_t = Z_t + \beta_1 Z_{t-1} + ... + \beta_q Z_{t-q}$$

Where X_t is the observation at time t, and Z_t is the white noise at time t. So the moving average process is influenced by weighted random effects from the past, the white noise up to lag q.
Using the backshift operator, the observations can be denoted as follows.

$$X_t = \theta_q(\boldsymbol{B})Z_t \tag{22}$$

With $\theta_q(\boldsymbol{B}) = 1 + \beta_1\boldsymbol{B} + ... + \beta_q\boldsymbol{B}^q$.
Note that an AR(1) process can be rewritten to a MA(∞) process in the following way

$$\begin{aligned}
&X_t = \alpha X_{t-1} + Z_t \\
&X_t = \alpha(\alpha X_{t-2} + Z_{t-1}) + Z_t \\
&... \\
&X_t = Z_t + \alpha Z_{t-1} + \alpha^2 Z_{t-2} + ...
\end{aligned}$$

where the condition for convergence is that $-1 < \alpha < 1$.
The moving average process has the following properties regarding to X_t, which follow directly from the properties of Z_t.

$$\mathbb{E}[X_t] = 0$$

$$Var(X_t) = \sigma_Z^2 \sum_{i-0}^{q} \beta_i^2$$

$$\rho(k) = \begin{cases} 1 & \text{for } k = 0 \\ \sum_{i=0}^{q-k} \beta_i\beta_{i+k} / \sum_{i=0}^{q} \beta_i^2 & \text{for } k = 1, ..., q \\ 0 & \text{for } k > q \\ \rho(-k) & \text{for } k < 0 \end{cases}$$

Where $\rho(k)$ is the autocorrelation function at lag k, which is defined for stationary processes in formula 2. Note that the autocorrelation function is not necessarily unique. Also note that there are no contributions for $k > q$, which can be used when estimating the value of order q using the autocorrelation function, because in the autocorrelation function a sudden drop should be seen after lag q, which indicates the value of order q. Another rule of thumb is that in the partial autocorrelation function, exponentially decaying behaviour can be seen after lag q.
The moving average process can be used to model a stationary time series. This means that the moving average process also needs to be stationary, and to obtain this, the process needs to be invertible. The process is invertible if $\theta_q(\boldsymbol{B}) \neq 0$, which is equivalent to the solutions of $\theta_q(\boldsymbol{B}) = 0$ being outside of the complex unit circle.

3.5.3 ARMA(p, q) model

The autoregressive moving average (ARMA) model is a combination of the autoregressive model and the moving average model and can be used to model stationary time series. The ARMA model is influenced by both levels and disturbances from the past. The ARMA model order (p, q) is denoted by the following.

$$X_t = (\alpha_1 X_{t-1} + ... + \alpha_p X_{t-p}) + (Z_t + \beta_1 Z_{t-1} + ... + \beta_q Z_{t-q})$$

It can also be formulated as follows, using the backshift operator.

$$\phi_p(\boldsymbol{B})X_t = \theta_q(\boldsymbol{B})Z_t \tag{23}$$

With $\phi_p(\boldsymbol{B}) = 1 - \alpha_1 \boldsymbol{B} - ... - \alpha_p \boldsymbol{B}^p$ and $\theta_q(\boldsymbol{B}) = 1 + \beta_1 \boldsymbol{B} + ... + \beta_q \boldsymbol{B}^q$.
To estimate the orders p and q, the autoregressive and moving average properties for the estimation can be used. This means that for the estimation of order p, the partial autocorrelation function can be analysed for a sudden drop and the autocorrelation function can be analysed for exponentially decaying behaviour. For the estimation of order q, the autocorrelation function can be analysed for a sudden drop and the partial autocorrelation function can be analysed for exponentially decaying behaviour.
Since the ARMA(p, q) process is modelling a stationary process, the invertibility and stationarity conditions of the moving average and autoregressive processes have to be satisfied. So it has to satisfy that $\phi_p(\boldsymbol{B}) \neq 0$ and $\theta_q(\boldsymbol{B}) \neq 0$.

3.5.4 ARIMA(p, d, q) model

The autoregressive integrated moving average (ARIMA) model can be used to model time series that contain trend. First the trend is removed using the finite differencing operator, which is defined as follows.

$$\nabla X_t = X_t - X_{t-1} = (1 - \boldsymbol{B})X_t$$

To remove a linear trend, the finite differencing has to be applied once, and for non-linear trends it has to be applied more than once, depending on the type. When the trend is removed, a stationary series remains, and the ARMA model can be applied. After applying the ARMA model, the series can be transformed back to include the trend again. Those transformations for the trend are automatically done in ARIMA models. The ARIMA model can be denoted as

$$\phi_p(\boldsymbol{B})(1 - \boldsymbol{B})^d X_t = \theta_q(\boldsymbol{B}) Z_t \tag{24}$$

Where d is the number of times finite differencing is applied.

3.5.5 SARIMA$(p, d, q)(P, D, Q)_s$ model

The seasonal autoregressive integrated moving average (SARIMA) model can be used for time series that contain both trend and seasonality. Using finite differencing and seasonal differencing, both the trend and seasonality can be corrected for to obtain a stationary time series. After applying the ARMA model, the series are transformed back. The seasonal differencing operator is defined as follows.

$$\nabla_s X_t = X_t - X_{t-s} = (1 - \boldsymbol{B}^s)X_t$$

The SARIMA model can be denoted as

$$\phi_p(\boldsymbol{B})\Phi_P(\boldsymbol{B}^s)(1 - \boldsymbol{B})^d(1 - \boldsymbol{B}^s)^D X_t = \theta_q(\boldsymbol{B})\Theta_Q(\boldsymbol{B}^s) Z_t \tag{25}$$

Where s is the seasonal period, and D is the number of times seasonal differencing is applied. The seasonal parts of the model $\Phi_P(\boldsymbol{B}^s)$ and $\Theta_Q(\boldsymbol{B}^s)$ are similar to the non-seasonal components, but involve backshifts of the seasonal period s.

3.5.6 ARMA Forecasting

To use an ARMA model to forecast future values of a time series, the following steps have to be taken. First the best fitting model has to be chosen based on the type of stochastic process (trend, seasonality). Next the parameter values have to be estimated, and the quality of multiple fitted models can be assessed by looking at the in-sample accuracy and goodness-of-fit measures. Then the best model can be chosen, and validated using independent data. When the model validation is completed, the model can be used to predict future values.
Forecasting of future values can be done by forward iteration. The one-step ahead prediction at time t is the following.

$$\hat{x}_t(1) = \mathbb{E}[X_{t+1} | X_t, X_{t-1}, ...]$$

The future values of Z are replaced by 0, and the present and past values of Z and X are replaced by their observed values, so the following forward iteration is obtained. By iterating the one-step ahead predictions, and using future predictions as future observations, predictions for multiple steps ahead can be calculated.

When the ARMA model is used in R, missing values are not a problem. The ARMA model uses the Kalman filter, that skips the update phase for missing values. This way no values have to be imputed after removing the zero values caused by holidays. This is preferable because an imputed value is not the correct observation and therefore can cause extra errors.

3.6 Model evaluation

For the evaluation and comparison of different models, it is important that every model is fit with the best amount of data. This can be different for models because the models may not have the same complexity. This is explained and performed in Section 4.5.
An indicator such as the mean squared error (MSE) is suitable for comparing different models and parameter combinations. Another option to compare models is to use Akaike's information criterion (AIC), which also takes into account the model complexity.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(x_t - \hat{x}_{t-1}(1))^2$$

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

Where L is the likelihood of the model, and k is the number of parameters that have been estimated.
Another way to evaluate if the best fitting model is chosen, is by observing the errors of the in-sample forecast. In-sample forecasting uses the same data for fitting and validation. The errors from this forecast indicate how much the forecast values differ from the observed values. If the model captures the level, trend and seasonality, all that is left is randomness that is normally distributed with mean zero and has no significant autocorrelations. If this is not the case, there is still some structure left in the time series that is not captured by the model.
A necessary property of the errors is that they need to have mean zero and no significant autocorrelations, so this has to be checked. The Ljung-Box test can be used to test if there is no significant autocorrelation left.
Once the forecasts are calculated, they can be used to calculate the moment that the storage will become empty or the moment the safety stock is reached. However this is the next step after the calculation of the forecasts so it is not necessary to take this into account at this point.

3.7 Non-negative forecasts

Some traditional forecasting methods are described, however in the methods it became clear that the traditional methods have one shortcoming. The traditional methods can produce forecasts that are negative, which is impossible when forecasting sales. First the sales are forecasted, and then the sales are used to update the inventory. This way negative sales forecasts will lead to an increase of inventory, which is impossible. To overcome this problem, two solutions are described in the following sections. Also a method is described to improve the forecasts and obtain non-negative forecasts at the same time.

3.7.1 Cut-off below zero

A simple way to prevent negative forecasts, is to use a cut-off below zero. This means that every negative forecast is set to zero, and this way no negative forecasts remain. An advantage of this approach is that it is very simple and it does not change the other forecasts. A disadvantage is that not all aspects of the data are taken into account because negative forecasts are produced and this indicates that a better modelling of the data might be possible.

3.7.2 Box-Cox transformation

Another way to ensure the forecasted values to be non-negative is to apply a logarithmic transformation. This way not the sales X will be modelled directly, but first the log-transformation $Y = \log(X)$ is applied, then the forecasts are calculated for Y and lastly the forecasts of Y are transformed back to forecasts of the sales X. A log-transformation is also called a Box-Cox transformation with $\lambda = 0$. The Box-Cox transformation is defined as

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda} \text{ if } \lambda \neq 0 \\ log(y) \text{ if } \lambda = 0 \end{cases} \tag{26}$$

Where y is an observation and $y(\lambda)$ is the transformed value of an observation. So in the case that $\lambda = 0$, the Box-Cox transformation is a log-transformation of the data points, where the logarithmic function is the natural logarithm.
To use the Box-Cox transformation with $\lambda = 0$, the data has to be strictly positive. When the data contains zero values, the two-parameter version of the Box-Cox transformation can be used, which allows a shift before the transformation, and is defined as follows [6].

$$y(\lambda) = \begin{cases} \frac{(y+\lambda_2)^{\lambda_1}1}{\lambda_1} \text{ if } \lambda_1 \neq 0 \\ log(y + \lambda_2) \text{ if } \lambda_1 = 0 \end{cases}$$

A common choice for the two-parameter transformation is $\lambda_1 = 0$ and $\lambda_2 = 1$, which has the property of mapping zero to zero [7]. A disadvantage of this choice is that forecasts can still become negative. A way to deal with zeros and have non-negative forecasts at the same time, is to choose $\lambda_1 = 0$ and to choose λ_2 as half of the smallest non-zero observation. This is common used in practice in case of detecting limits in measurement systems.
A disadvantage of the Box-Cox transformation is that the forecast value on the original scale is an estimation of the median, but in most cases the quantity of interest is the mean, instead of the median. When the distribution of the forecast is symmetric, the mean and median are almost equal, but when the distribution is skewed, this can lead to a bias in the forecasts. Taylor [8] described an approximation method that can be used to estimate the mean on the original scale, which is based on a Taylor expansion around the mean. This procedure is integrated in R and can be used to estimate the mean after the Box-Cox transformation.

3.7.3 Forecasting with bootstrapping and Box-Cox transformation

A study of Bergmeir, Hyndman and Benitez [5] combined multiple existing methods to obtain better forecasting results than using a traditional method only. The idea in this study is to combine exponential smoothing, the Box-Cox method, time series decomposition and bootstrapping of the residuals. The classical additive exponential smoothing as described in Formula 13 and the classical Box-Cox as described in Formula 26 will be used. Time series decomposition will be used according to the STL method as described in Section 3.3.1. The last method that will be used is bootstrapping of the residuals. Bootstrapping in general means generating data samples from the original sample using replacements. Bootstrapping is a method that can be used to improve the forecast accuracy. When applying the method, B time series are generated which are similar to the original observed series, where B is a number that can be chosen by the user. Then for all the generated series, a forecast is produced, and lastly those forecasts are pointwise averaged to obtain one forecast that is more accurate than the forecast obtained by only using the classical exponential smoothing method.
In this case, the replacements only occur in the residuals of the time series decomposition. Residual series are constructed by random sampling with replacement values from the observed residuals. Those residuals series are put together with the other series components and this way similar series as the original series are constructed.
There may be autocorrelation present in the residuals of a time series decomposition, so one can not simply replace residuals, but a 'blocked bootstrap' is used. In a blocked bootstrap, consecutive sections of the residuals of the time series are selected at random and joined together to preserve local characteristics like autocorrelation. The length of the blocks is usually 2 times the seasonality period, so for example for a weekly pattern in daily data, the blocks are of length 14.

Aggregated bootstrapping, also called bagging, is the method where a forecast is produced for all B bootstrapped series, and then the average of the forecasts is calculated to obtain one forecast for the original time series.

All together, those methods are combined to obtain an overall procedure. This procedure has the following steps. First the value of λ for the Box-Cox transformation is estimated, and the corresponding Box-Cox transformation is applied. Next the time series decomposition according to the STL method is applied, which results in a trend, a seasonal and a residual component. Then the residuals are bootstrapped using the moving block bootstrapping (MBB) method. Then the trend, seasonal and bootstrapped residual components are added together again and the Box-Cox transformation is inverted. This way B bootstrapped time series on the original scale are obtained. For those series, exponential smoothing forecasts are calculated and those can be combined to obtain one forecast for the original time series. This procedure is summarized in the following algorithm.

Algorithm 1 Procedure Holt-Winters with bootstrapping

```
Estimate λ for Box-Cox
Apply Box-Cox with λ
Apply STL to obtain seasonal, trend, residuals
Bootstrap residuals with MBB
Combine seasonal, trend, bootstrapped residuals
Invert Box-Cox
Apply additive Holt-Winters method
Combine forecasts on bootstrapped series into one forecast
return Forecast
```

When applying bagging and ES, variance reduction happens and therefore the forecast errors become smaller and the forecasts become better. The results of the study [5] show that the method outperforms the classical exponential smoothing method consistently based on the mean absolute scaled error (MASE). In Section 4.6 it will be seen if this method also performs better for the data in this project. A disadvantage is that this method is a lot slower than classical exponential smoothing because more computation has to be done.

The method can not be applied on data with zero-values, because the classical Box-Cox transformation does not work for zero-values. Also, the method described above can predict values below zero, because the forecasts are calculated after the inverse Box-Cox transformation is applied, which is not desired for the data in this report. A possibility to be able to model non-negative data containing zero-values is to use the two-parameter Box-Cox transformation. After the Box-Cox transformation, the STL decomposition and the bootstrapping, the exponential smoothing forecasting can be applied and the results can be transformed back by applying the inverse Box-Cox transformation to get results on the original scale. Lastly the forecasts on the original scale of the bootstrapped series are averaged to get one forecast. This adapted procedure is summarized in the following algorithm.

Algorithm 2 Procedure Holt-Winters with bootstrapping adapted for zero-values and non-zero forecasts

```
Apply two-parameter Box-Cox with λ = 0
Apply STL to obtain seasonal, trend, residuals
Bootstrap residuals with MBB
Combine seasonal, trend, bootstrapped residuals
Apply additive Holt-Winters method
Invert Box-Cox
Combine forecasts on bootstrapped series into one forecast
return Forecast
```

This way zero-values are not a problem and there will be no forecasts below zero. However, it is not shown that this also leads to improved results compared to the classical exponential smoothing method, but this solves most of the problems that occurred.

Oliveira [9] investigated the bootstrapping method applied to the SARIMA method, and this can also be used to improve the forecasts. The same procedure as in Algorithm 1 and Algorithm 2 can be used where the step: 'Apply additive Holt-Winters method' will be replaced with the step: 'Apply the SARIMA method'.

4 Results

In this chapter the methods from the literature research will be applied on data of Bottomline. There is a lot of data available, and choices have to be made about what data to use. The goal is to improve the forecasts for the inventory of the storages. To be able to improve the forecasts in general, and not only for one particular storage, multiple storages with different properties have to be analysed. Also data of different periods in history will be used to make sure the forecasting method is suitable for all situations.
In the selection process of the storages, it is preferable that there is an inventory measurement (so a data point) at least once a day, because otherwise the weekly pattern set by the company has too much influence. Missing measurements will not lead to missing data in the time series due to the weekly pattern of the company that is used, but the results might become less precise if this is used a lot. For fast selling storages this is not a problem in most cases, but especially for slow selling storages that get only a few deliveries per year, there is no daily data available.
The storages are ordered by selling speed, and two fast selling storages (respectively storage 1 and 3), two medium speed selling storages (storage 2 and 5) and a slow selling storage (storage 4) are analysed in the following sections. There will be looked at two time periods, which are 2019 and 2020, where from 2020 only 20 weeks of data are used.

4.1 Exploratory Data Analysis

The first step is to explore the data and know more about the properties of the data.

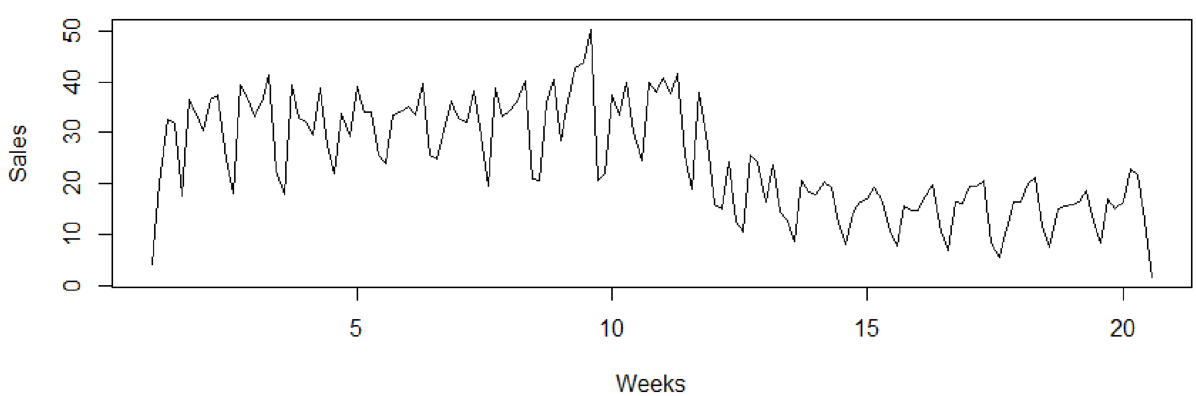

Figure 4: Time sequence plot 2020 storage 1

In Figure 4, we show the time sequence plot of storage 1 (a fast selling storage), relative to the first 20 weeks of 2020. The x-axis is in days, where we can see the weekly pattern of 7 days. The x-axis is labelled weekly, where the label of the week itself is in the middle of the week. The figure shows that the sales decay over

time, which is probably due to the corona virus. This decay over time might indicate the presence of a trend component in the data, but it can also be a sudden drop which is hard to model. It can also be seen that there is a repeating pattern in the sales, the seasonality, where the frequency is 7 days, so it is a weekly pattern.
When the time sequence plot is observed, it can be seen that the amplitude of the seasonality component (the difference between the local minima and maxima of one week) is larger when the level is higher, but the difference is very small. This indicates that the seasonality might be multiplicative. For more certainty, this can be investigated further using time series decomposition.
In Figure 4 it can be seen that in week 9 there is one sales value that is higher than the others, this might be an outlier.

Another aspect of the exploratory data analysis is the analysis of the autocorrelation function and the partial autocorrelation function. In the figures below the time sequence plot, the autocorrelation plot and the partial autocorrelation plots of storages 1 for 2019 and storage 2 for 2020 are shown. Figure 5 refers to the same data shown in Figure 4.

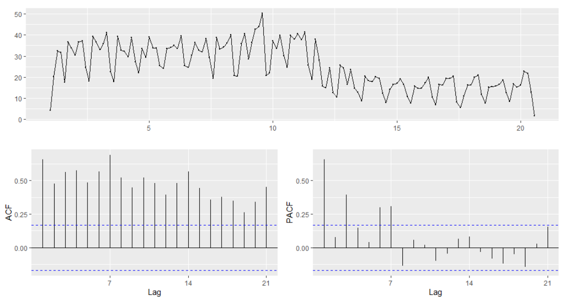

Figure 5: Storage 1 2020

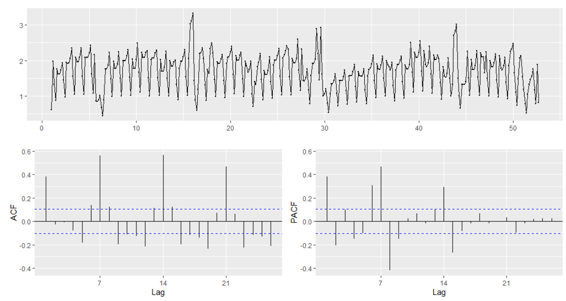

Figure 6: Storage 2 2019

In the autocorrelation function plots, there are significant contributions at lag 7, 14 and 21. In the partial autocorrelation plots, significant contributions can be seen at lag 7. This indicates that there is a seasonality contribution, and the seasonality period equals 7 days, as was already noticed looking at the time sequence plot. The partial autocorrelation function plot does not have significant contributions for lag 14 and 21 in most of the cases. This indicates that there is only one seasonality period, which is 7, and the significant contributions at lag 14 and 21 in the autocorrelation function are due to the indirect correlation.
For some time periods, for example in Figure 5, it can be seen in the autocorrelation function plot that in between the seasonality contributions at multiples of 7, there are also significant contributions, for example lags 2, 3, 4, 5 and 6. This indicates there can be contribution of trend, and can be due to the corona virus as mentioned before.

In the figures below, the time series decompositions are shown. Figures 7 and 8 refer to the same data as used in Figure 6, and Figures 9 and 10 refer to the same data as used in Figure 5.
In the decomposition figures, the upper plot is the original time series. The second plot displays the trend component. The seasonality component is visualized in the third plot from the top, and in the last plot the residuals are shown.

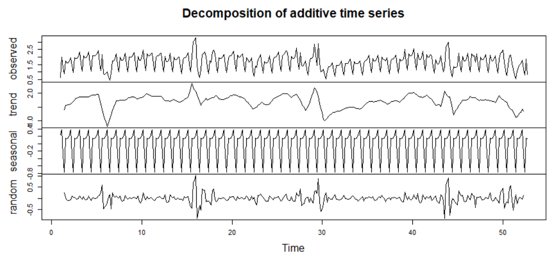

Figure 7: 2019 storage 2 decomposition in case of additive seasonality

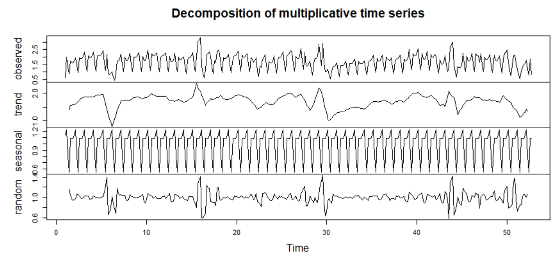

Figure 8: 2019 storage 2 decomposition in case of multiplicative seasonality

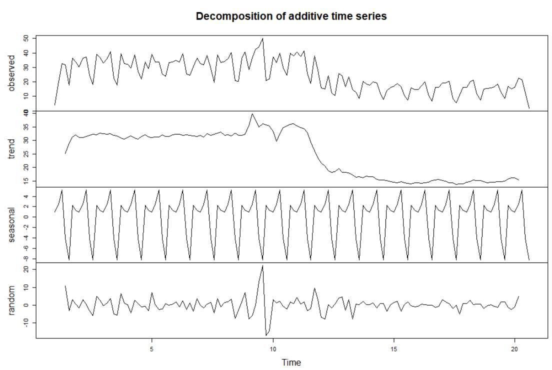

Figure 9: 2020 storage 1 decomposition in case of additive seasonality

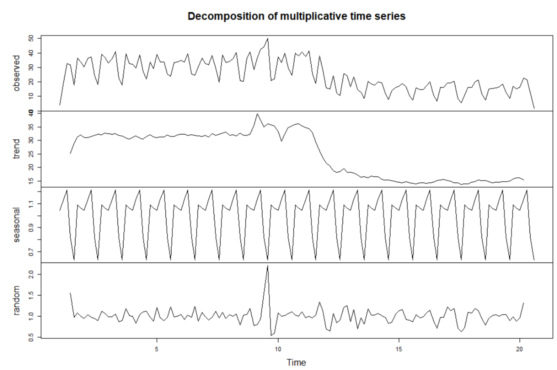

Figure 10: 2020 storage 1 decomposition in case of multiplicative seasonality

In the figures on the left side, the decomposition is applied assuming additive seasonality and trend, and in the figures on the right side, the decomposition is applied assuming multiplicative seasonality and trend. A multiplicative model causes that the amplitude of the seasonality component gets larger as the level gets larger. In the figures it can be seen that there are no big differences between the random terms of the additive decomposition and the random terms of the multiplicative decomposition, except that the random terms of the additive model are centered around zero and the random terms of the multiplicative model are centered around 1. This is because of the difference in calculation between the additive and multiplicative model. The small differences can be caused by the fact that there is no very clear trend in the data of 2019, which makes it hard to identify the difference between the additive and the multiplicative model. In the data of 2020 shown in Figures 9 and 10 the trend is more clear, and there it can be seen that starting at week 13 the random terms are bigger for the multiplicative model than for the additive model (relatively bigger, because the additive random terms are centered around zero and the multiplicative random terms are centered around one).
In figures 7 and 8 every 10 to 15 weeks, bigger spikes can be seen in the residuals. Those bigger spikes are caused by bigger observed values at those same days, and can be due to for example holidays.
The figures below show an analysis of the residuals of the decompositions of storage 1 in 2020. In both figures, the upper figure is the time sequence plot of the residuals. The figure on the left bottom is the autocorrelation function, and the figure on the right bottom is a histogram of the residuals and the red line draws a normal distribution.

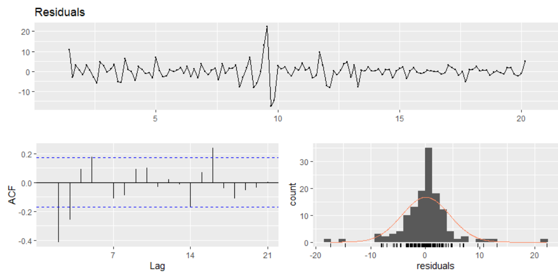

Figure 11: 2020 storage 1 residuals additive decomposition

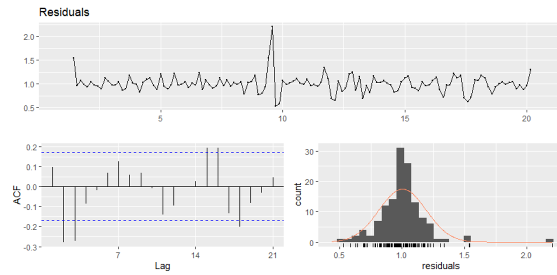

Figure 12: 2020 storage 1 residuals multiplicative decomposition

In Figure 11 and 12 it can be seen that for both models, there are still significant autocorrelations left. In the histograms it can be seen that the shape of the residuals looks symmetric. For both models, the residuals have still autocorrelations left and they are not normally distributed, so with respect to those properties there is no preference for one of the models.
Another aspect that can be useful in the choice between an additive and a multiplicative model is if there are zeros in the data, because some multiplicative models can not deal with zero values in the data (for example multiplicative Holt-Winters exponential smoothing). Although the zero values caused by holidays that were set with knowledge are removed, there can still be zero values for example in storages that are closed on weekend days. It is preferable not to remove the structural zero values because this will make the approach more complicated as not every storage has the same week-length and seasonality period, and this can also lead to confusion. Also as the zero values are structural, an accurate forecast can be made for the zero values.
To be able to use a multiplicative model when the data contains zero values, a solution can be to add a constant value to all observations, and subtract the same constant value from all predictions, but this makes the forecasts less reliable, and adds extra uncertainty in choosing the constant value that will be used. This is a disadvantage of the multiplicative model, and therefore the additive model is chosen and this will be used from here on.

For storage 1 in 2020, both contributions of trend and seasonality are present, which was seen in the autocorrelation function in figure 5. To obtain a stationary time series, one time finite differencing is applied to remove trend. Then the time sequence plot, the autocorrelation and the partial autocorrelation plots for the so obtained time series are shown.

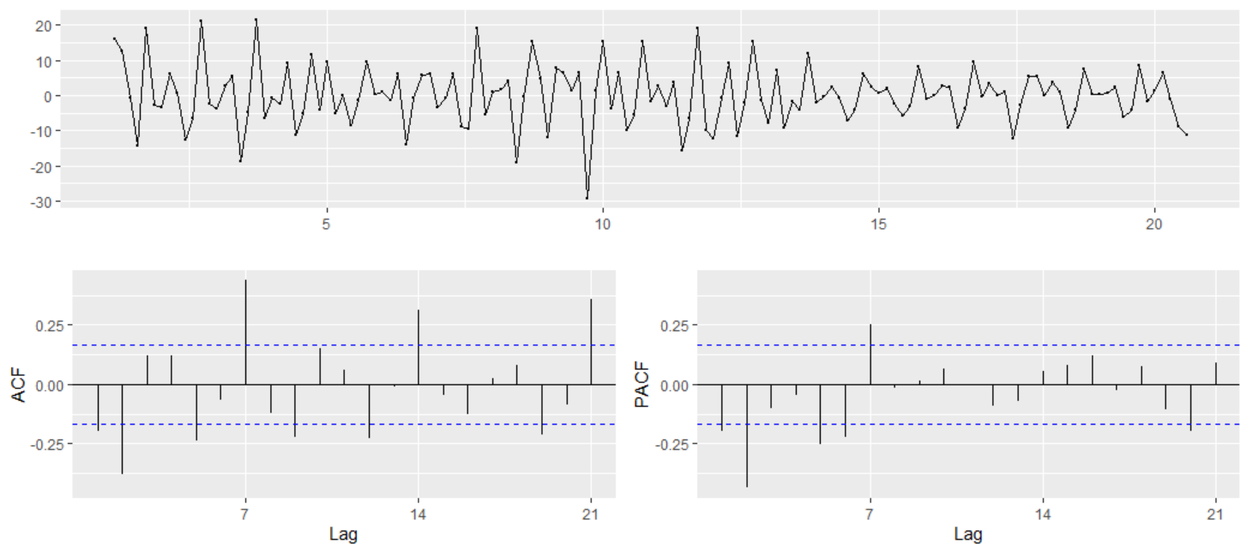

Figure 13: One time finite differencing 2020 storage 1

In Figure 13 it can be seen that indeed the trend has been removed because there is no significant contribution for lag 1. Next seasonal differencing for lag 7 can be applied to remove the weekly seasonality pattern.

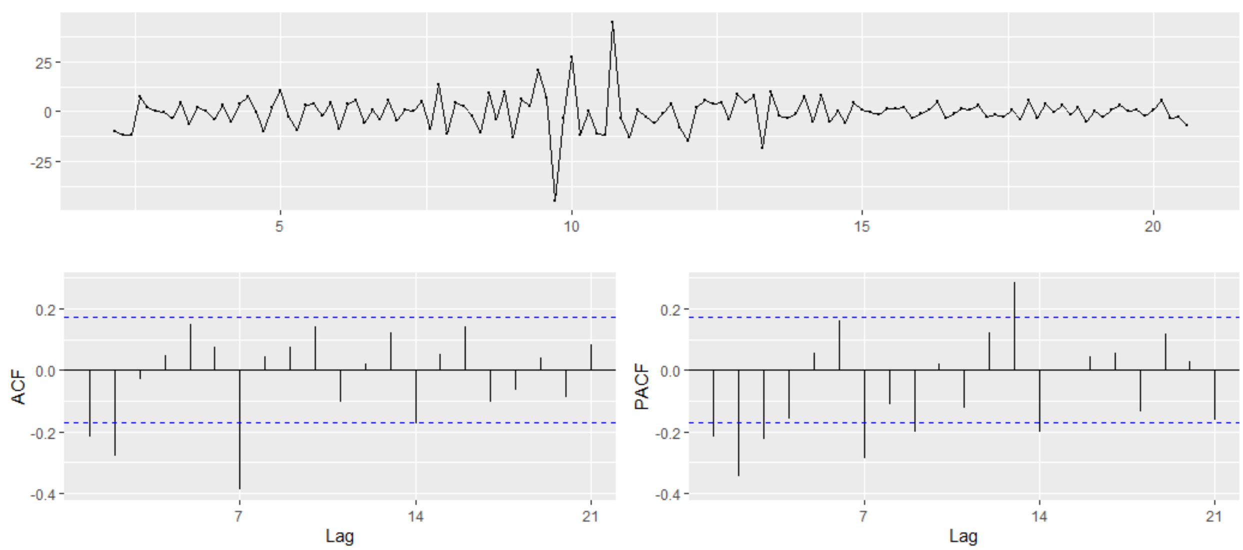

Figure 14: One time finite differencing and one time seasonal differencing 2020 storage 1

Figure 14 above displays the time series after one time finite differencing and one time seasonal differencing for a seasonality period of 7 days. In the time sequence plot it can be seen that there are no clear trend and seasonality left. There are still some autocorrelations left, but those can not be removed by applying finite or seasonal differencing again, which can be seen in the figures below. As long as there are no significant autocorrelations left in the residuals after fitting the forecasting model, this is not a problem.

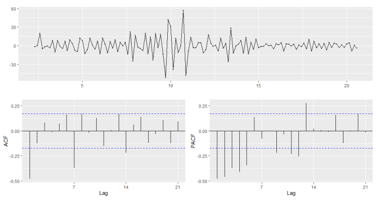

Figure 15: Two times finite differencing and one time seasonal differencing 2020 storage 1

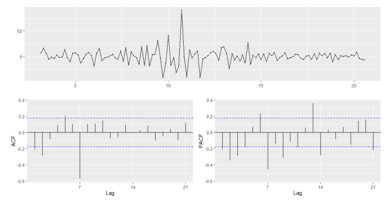

Figure 16: One time finite differencing and two times seasonal differencing 2020 storage 1

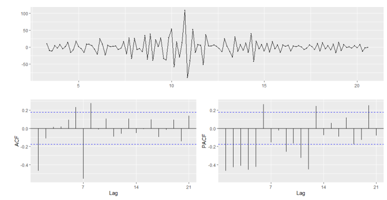

Figure 17: Two times finite differencing and two times seasonal differencing 2020 storage 1

It is preferable that the results are generally applicable, and to achieve this, exploratory data analysis is also applied to storage 2, 3, 4 and 5, which can be found in the appendix A.1.
All together, seasonality is present for all storages. For most storages there is trend present, and in some cases where not much trend is present, the model can estimate the trend effect with zero. Also, when trend is present and it is not modelled, it will cause biased residual terms.
Therefore both trend and seasonality are considered in the modelling of the storages. The storages will be modelled with an additive model with trend and seasonality components, and it is important to use long periods of data (at least more than two weeks, but preferably more) to be able to estimate the trend and seasonality components.

4.2 Exponential smoothing models

The conclusion from the exploratory data analysis is that the time series will be modelled with additive trend and seasonality. Therefore the Holt-Winters exponential smoothing method with additive seasonality and trend will be used as described in Formula 13 and initial values described in Formulas (15-17). The method will be applied in R using the $HoltWinters$ function from the $fpp2$ package [2].

First we will look at the results for storage 1 in 2020. After performing the Holt-Winters exponential smoothing method in R, the following smoothing parameter estimates are found.

$$\hat{\alpha} = 0.2588404$$
$$\hat{\beta} = 0.05184603$$
$$\hat{\gamma} = 0.1216038$$

All smoothing parameters are close to zero, which means that more weight is given to observations further in the past, and the model is called slow learning. The estimates for the level, trend and seasonal components are the following for the last time step T, which is the day of the last data point that is used to fit the model.

$$L_T = 15.43456573$$

$$T_T = -0.04984005$$

$$S_T = \begin{bmatrix} S_{T-s} \\ S_{T-s+1} \\ \vdots \\ S_T \end{bmatrix} = \begin{bmatrix} 1.22070877 \\ 0.65989172 \\ 0.18909180 \\ 2.63909848 \\ 4.21420170 \\ -4.29873911 \\ -9.50064755 \end{bmatrix}$$

Where S_{T-s} is the seasonal component for Mondays, and so on.
In the figure below, the in-sample forecast can be seen. This is the forecast until time T (in red), plotted in one figure together with the observed values (in black). Note that the same data is used for both fitting and validation. When comparing different models in Section 4.6, also the out of sample accuracy will be

analysed, i.e. the accuracy of the model in predicting observations outside the range of time used for fitting the model.

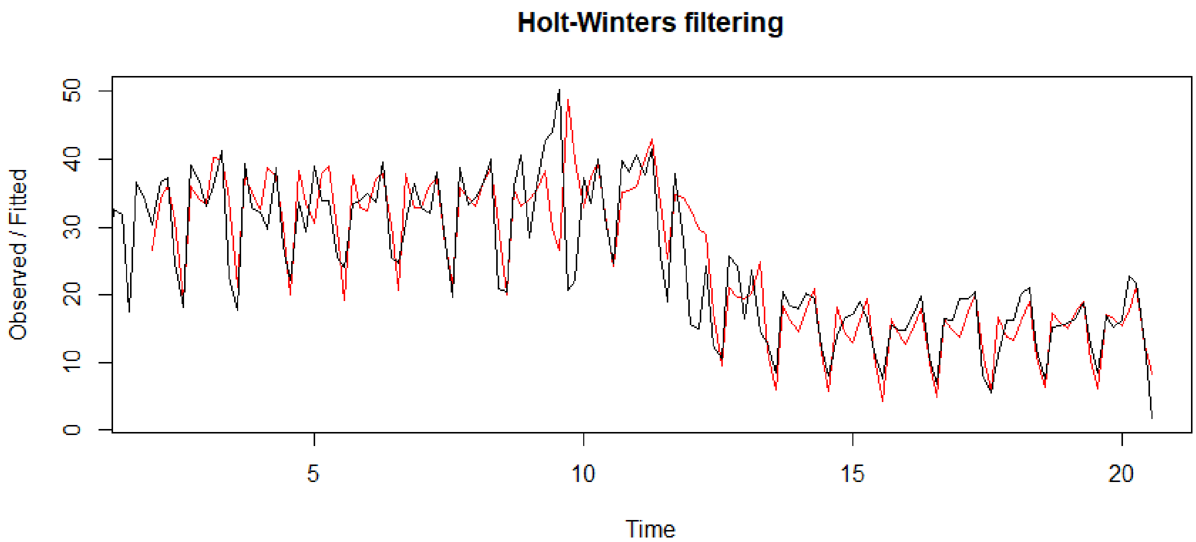

Figure 18: In-sample Holt-Winters forecast for 2020 storage 1

In Figure 18 it can be seen that the forecast successfully captures the seasonal pattern, and it also captures the trend. The additive seasonality can be recognised in the forecast, because the seasonality amplitude stays similar over time.
It is interesting to look at the residuals of the in-sample forecast. The residuals are the differences between the observations and the fitted values. The time sequence plot, autocorrelation function and partial autocorrelation function are shown below.

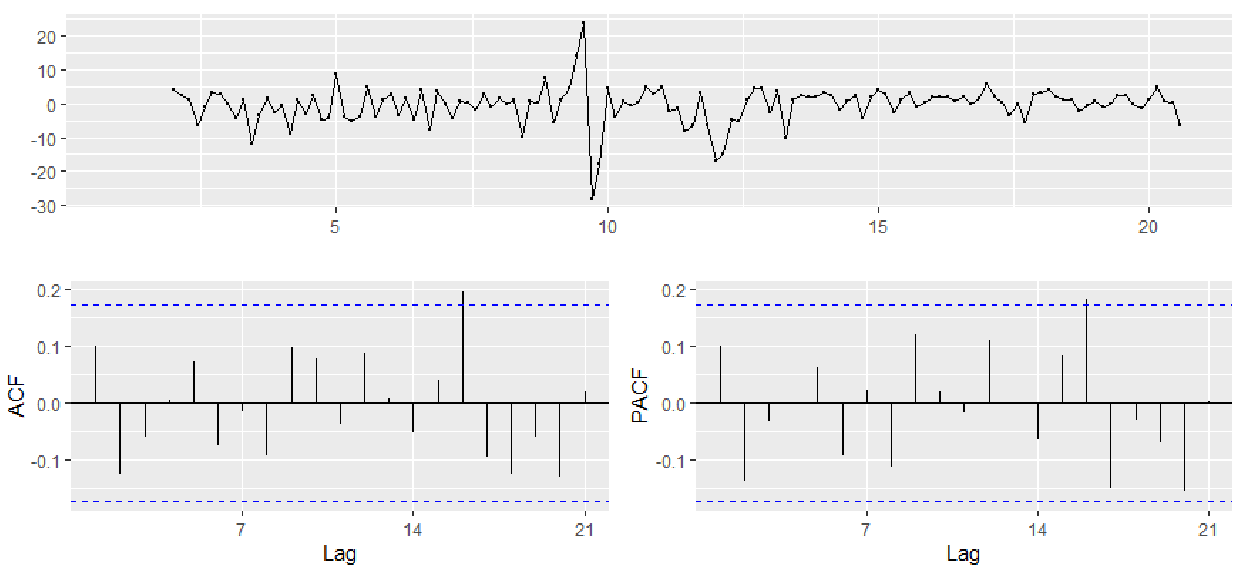

Figure 19: Residuals of in-sample Holt-Winters forecast 2020 storage 1

The residuals are centered around zero. The autocorrelation for lag 16 is a little above the blue significance

line, so this could indicate that the residuals are not totally random and that some structure is left. The Ljung-Box test can be performed to check if there is overall significant autocorrelation left, with significance level $\alpha = 0.05$. The p-value of the Ljung-Box test equals 0.331, which is bigger than α. This means that the null-hypothesis, that there is no significant autocorrelation left, is not rejected.
Another aspect that has to be checked, is if the residuals are normally distributed. This can be checked using the Shapiro-Wilk test, with significance level $\alpha = 0.05$. The null-hypothesis of the Shapiro-Wilk test is that the data is normally distributed. when applying the test, an p-value of $4.998 \cdot 10^{-9}$ is found. This means that p is smaller than α, and the null hypothesis is rejected, so it is not assumed that the residuals are normally distributed. This might cause errors in the confidence intervals.
To get more insight in the distribution of the residuals, a histogram is visualized in Figure 20. In the histogram it can be seen that the residuals are not normally distributed. The values are close to zero and there the frequency of values further from zero is decaying, and the residuals are a bit skewed to the right.

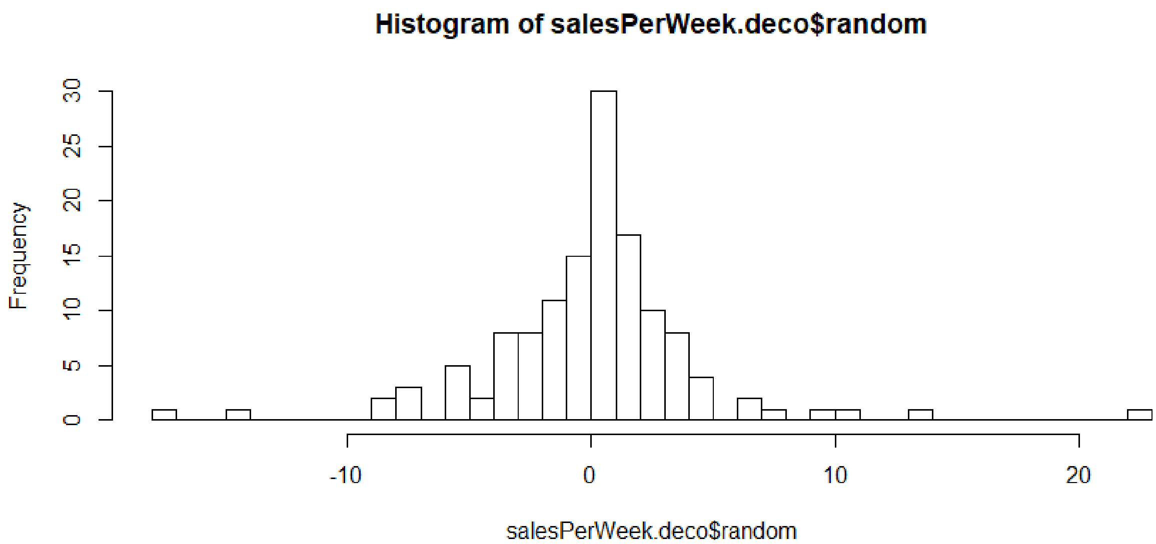

Figure 20: Histogram of residuals Holt-Winters forecast 2020 storage 1

In the figure below, the forecast for 3 weeks in the future is visualized.

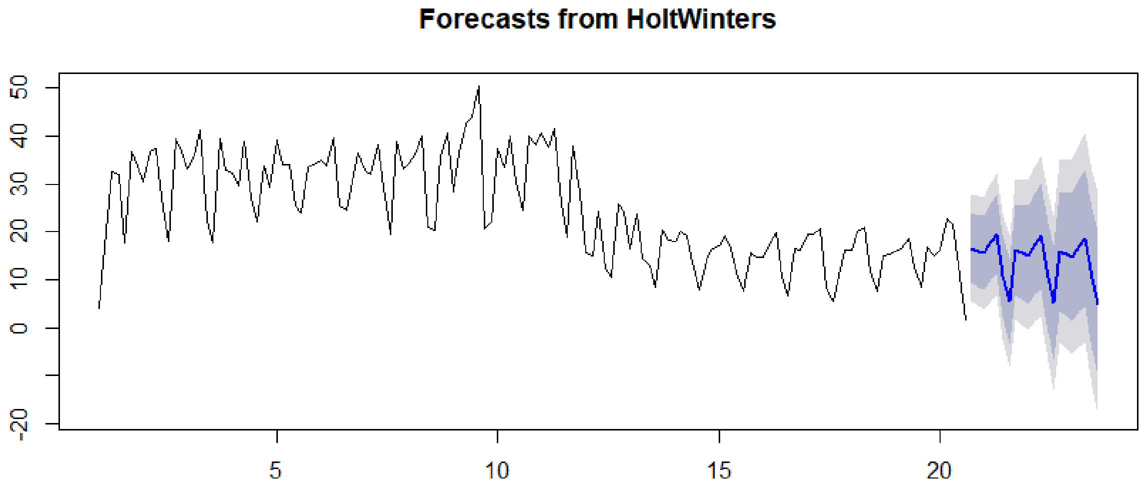

Figure 21: Holt-Winters forecast 2020 storage 1

The blue line in the figure represents the forecasted values. The dark grey area is the 80% prediction interval for the forecast, and the light grey area is the 95% prediction interval for the forecast. In the forecast, the seasonality pattern can be seen, and there seems to be a small decrease in the mean.
In addition to the exponential smoothing application to storage 1, a similar application is also done for storage 2. This can be found in Appendix A.2
To conclude, the Holt-Winters model works well for some storages, but for other storages there is still correlation left in the residuals, which means that this might not be the best model. The out-of-sample accuracy will be evaluated in Section 4.6 to see the performance of the model compared to real observations. Then the model will also be compared to other models to see the differences in performance between different models.

4.2.1 Holt-Winters method with cut-off at zero

In Figures 22 and 23, the Holt-Winters forecasts for storage 1 and 4 are visualised where the forecasts are cut-off below zero, which means that all negative forecasts are set to zero.

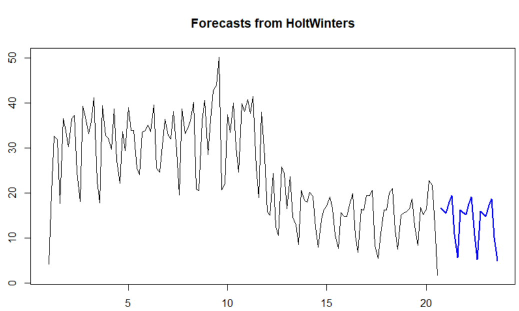

Figure 22: Holt-Winters forecast with cut-off at zero for 2020 storage 1

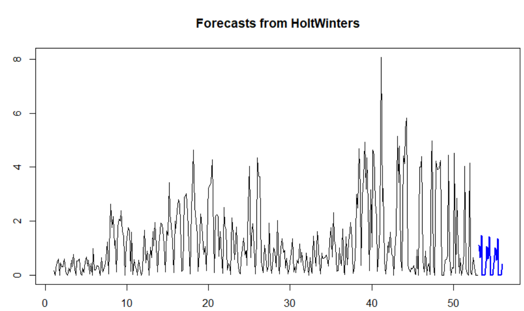

Figure 23: Holt-Winters forecast with cut-off at zero for 2019 storage 4

It can be seen that the forecasts are the same as for the classic Holt-Winters method, but the values below zero are all set to zero. An advantage of this method is that the other forecasts remain the same, especially for storages that do not have sales close to zero, nothing changes in the forecasts. A disadvantage is that the forecasts have to be corrected after the application of the forecasting method, which means that the patterns and constraints are not fully captured by the model.

4.2.2 Holt-Winters method with Box-Cox transformation

The data that is used in this project is non-negative, and it has been seen that in some cases, the data contains zero values. In the exploratory data analysis in section 4.1 is concluded that the time series will be modelled with additive trend and seasonality. Therefore the Holt-Winters exponential smoothing method with additive seasonality and trend will be applied in combination with a two-parameter Box-Cox transformation with $\lambda_1 = 0$ and λ_2 is half of the smallest non-zero value in the data.
First we will look at the results for storage 4 in 2019. After performing the Box-Cox transformation with $\lambda_2 = 0.0000275517906567131 \cdot 10^3$ liter, and the additive Holt-Winters exponential smoothing method in R, the following smoothing parameters are found.

$$\hat{\alpha} = 0.2003821$$
$$\hat{\beta} = 0.01228698$$
$$\hat{\gamma} = 0.1300389$$

All smoothing parameters are close to zero, which means that more weight is given to observations further in the past, and the model is called slow learning. The estimates for the level, trend and seasonal components are the following for the last time step T.

$$L_T = -4.3285474$$
$$T_T = -0.0372146$$
$$S_T = \begin{bmatrix} S_{T-s} \\ S_{T-s+1} \\ \vdots \\ S_T \end{bmatrix} = \begin{bmatrix} 1.3239506 \\ 1.4479016 \\ 1.0410509 \\ 1.8408850 \\ 0.2392592 \\ -2.9428124 \\ -0.9618652 \end{bmatrix}$$

In the figures below, the in-sample forecast and the residuals of the in-sample forecast of the Box-Cox transformed time series can be seen. The in-sample forecast is shown in red and the observed values are shown in black. Note that the same data is used for both fitting and validation. When comparing different models in Section 4.6, also the out of sample accuracy will be analysed.

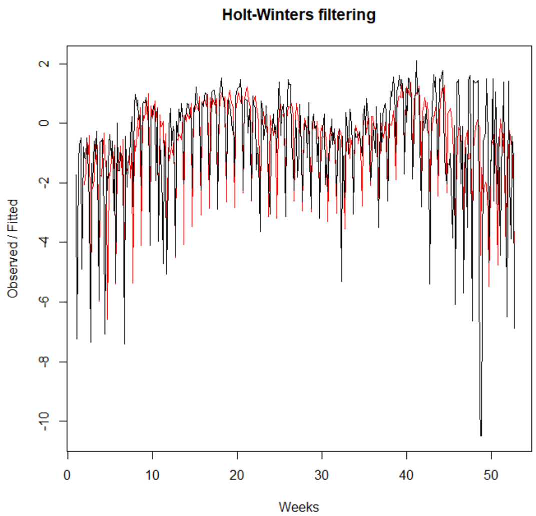

Figure 24: In-sample forecast for 2019 storage 4 on transformed scale

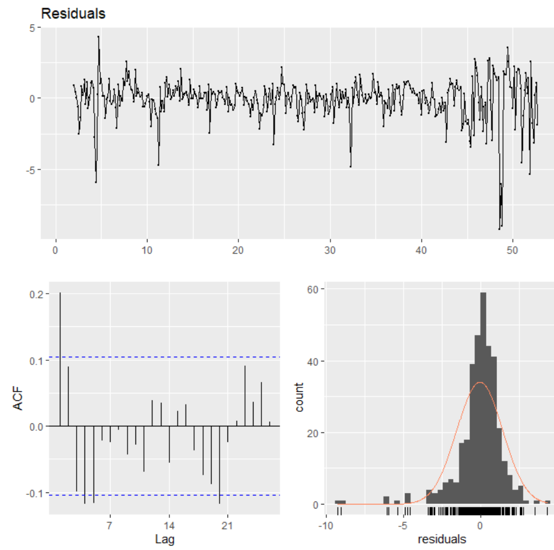

Figure 25: Residuals of in-sample forecast 2019 storage 4

Where S_{T-s} is the seasonal component for Mondays, and so on.
In Figure 24 it can be seen that the forecast successfully captures the seasonal pattern, and it also captures the trend. The additive seasonality can be recognised in the forecast, because the seasonality amplitude stays similar over time. The time sequence plot, autocorrelation function and a histogram of the residuals are shown in Figure 25. The residuals are centered around zero. The autocorrelation for lag 16 is a little above the blue significance line, so this could indicate that the residuals are not totally random and that some structure is left. The Ljung-Box test can be performed to check if there is overall significant autocorrelation left, with significance level $\alpha = 0.05$. The p-value of the Ljung-Box test equals 0.0003577, which is smaller than α. This means that the null-hypothesis, that there is no significant autocorrelation left, is rejected. So for all three storages analysed here, there are significant autocorrelations left, which means that the model can be improved.
Another aspect that has to be checked, is if the residuals are normally distributed. This can be checked using the Shapiro-Wilk test, with significance level $\alpha = 0.05$. The null-hypothesis of the Shapiro-Wilk test is that the data is normally distributed. when applying the test, an p-value of $2.2 \cdot 10^{-16}$ is found. This means that p is smaller than α, and the null hypothesis is rejected, so it is not assumed that the residuals are normally distributed. Fortunately, it is not necessary that the residuals are normally distributed, so the model can still be used. In the histogram it can be seen that the residuals are not normally distributed, but there is a centering around zero.

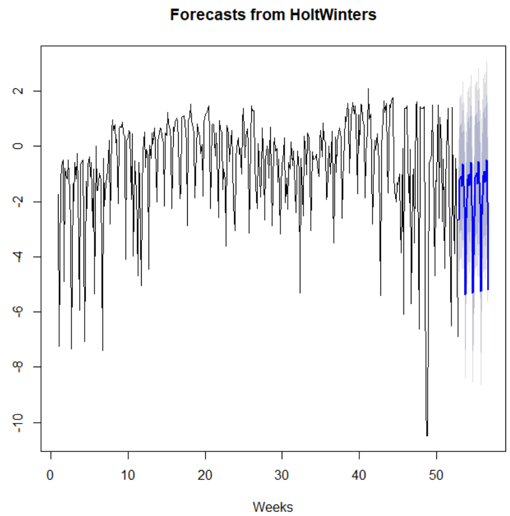

Figure 26: Holt-Winters forecast 2019 storage 4 on transformed scale

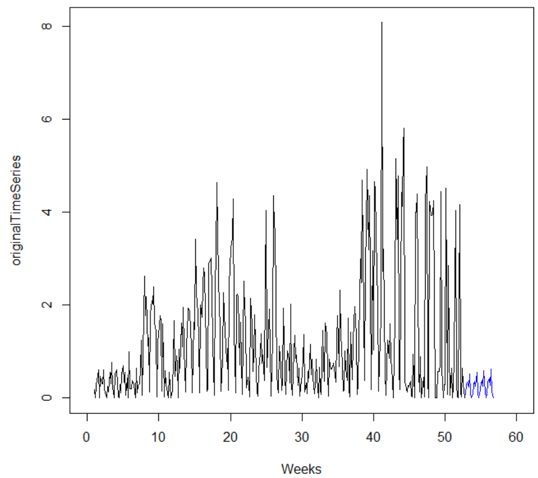

Figure 27: Holt-Winters forecast 2019 storage 4 on original scale

In Figure 26 the Holt-Winters forecast in the Box-Cox transformed scale can be seen. The blue line in the figure represents the forecasted values. To dark grey area is the 80% prediction interval for the forecast, and the light grey area is the 95% prediction interval for the forecast. In the forecast, the seasonality pattern can be seen, and there seems to be a small trend in the forecast.
In Figure 27 the forecast in the original scale can be seen, so this is the forecast after the inverse Box-Cox transformation is applied. It can be seen that the forecasts look very different, and the forecasts on the original scale seem to be a lot smaller than the observations before. In this time series, zero-values are present and it can be seen that indeed the forecasts are non-negative.

A similar analysis is done for storage 1 and 2, which can be found in Appendix A.4.
All together, a forecast that is based on a Box-Cox transformation combined with the Holt-Winters model produces non-negative forecasts. However the forecasts overall seem to be less accurate, and this will be investigated more in Section 4.6 where the out-of-sample evaluation will be performed.

4.2.3 Holt-Winters method using bootstrapping and Box-Cox transformation

Another way to produce forecasts that are non-negative is by using a Box-Cox transformation in combination with the Holt-Winter method and bootstrapping. Again for the Box-Cox transformation $\lambda_1 = 0$ is chosen and $\lambda_2 = 0.835409 \cdot 10^3$ liter, which is half of the smallest non-zero value in the data.
The procedure is performed in R, where the moving block bootstrap is performed using the *bld.mbb.bootstrap* function from the $forecast$ package [2].

First we will look at the results for storage 1 in 2020. Due to the bootstrapping, multiple series are constructed and fitted, and therefore not one Holt-Winters model can be described.

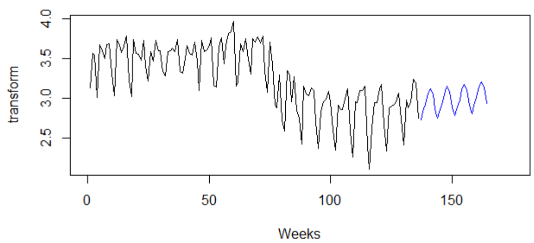

Figure 28: Holt-Winters forecast with bootstrapping for 2020 storage 1, transformed scale

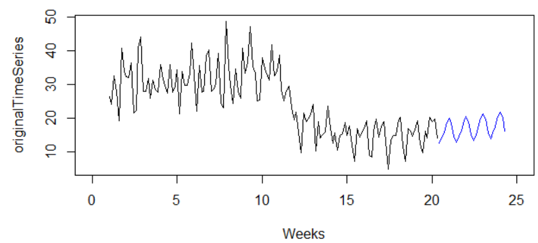

Figure 29: Holt-Winters forecast with bootstrapping for 2020 storage 1, original scale

In Figure 28 the Holt-Winters forecast with bootstrapping can be seen in the Box-Cox transformed scale. In the forecast, the seasonality pattern can be seen, and there seems to be a small trend in the forecast. In Figure 29 the forecast in the original scale can be seen, so this is the forecast after the inverse Box-Cox transformation is applied. In Section 4.6 the forecasts will be compared to observations.

A similar analysis is done for 4 in 2019, which contains zero values. For the Box-Cox transformation again $\lambda_1 = 0$ and $\lambda_2 = 0.0000275517906567131 \cdot 10^3$ liter.

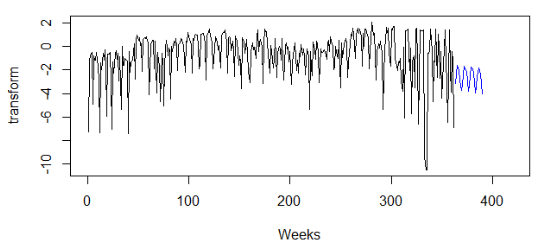

Figure 30: Holt-Winters forecast with bootstrapping for 2019 storage 4, transformed scale

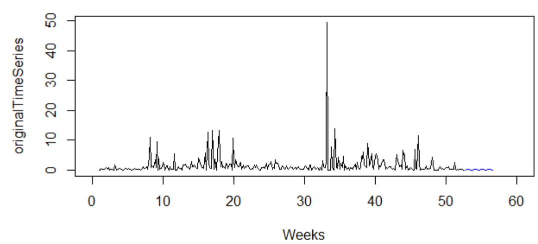

Figure 31: Holt-Winters forecast with bootstrapping for 2019 storage 4, original scale

In Figure 30 the Holt-Winters forecast with bootstrapping can be seen in the Box-Cox transformed scale. In the forecast, the seasonality pattern can be seen, and there seems to be a small trend in the forecast. On sight the forecasts seem to be quite low compared to the historic observations. In Figure 31 the forecast in the original scale can be seen. It can be seen that there are no negative forecasts, and the pattern of the zero values seems to be accurate. However, the forecasts seem to be very low compared to the historic observations.

All together, a forecast that is based on a Box-Cox transformation combined with the Holt-Winters model with bootstrapping produces non-negative forecasts. However the forecasts overall seem to be less accurate, and this will be investigated more in Section 4.6 where the out-of-sample evaluation will be performed.

4.3 ARMA models

In the exploratory data analysis was concluded that the time series contains both trend and seasonality. Therefore the SARIMA$(p, d, q)(P, D, Q)_s$ model will be used. In order to find AR and MA order for the model, the autocorrelation function and partial autocorrelation functions can be analysed.

In the exploratory data analysis it became clear that to remove the trend and seasonality using finite and seasonal differencing, both differencing methods have to be applied once. Therefore differencing parameters d and D will both be set to 1.

For the choice of the order of autoregression p, the partial autocorrelation function can be used. Because

a SARIMA process is used now, there are also other contributions for the partial autocorrelation function, but the function can still be used for the estimation of parameter p. For the choice of the moving average order q, the autocorrelation function can be used.
First the autocorrelation and partial autocorrelation functions of storage 1 in 2020 will be analysed.

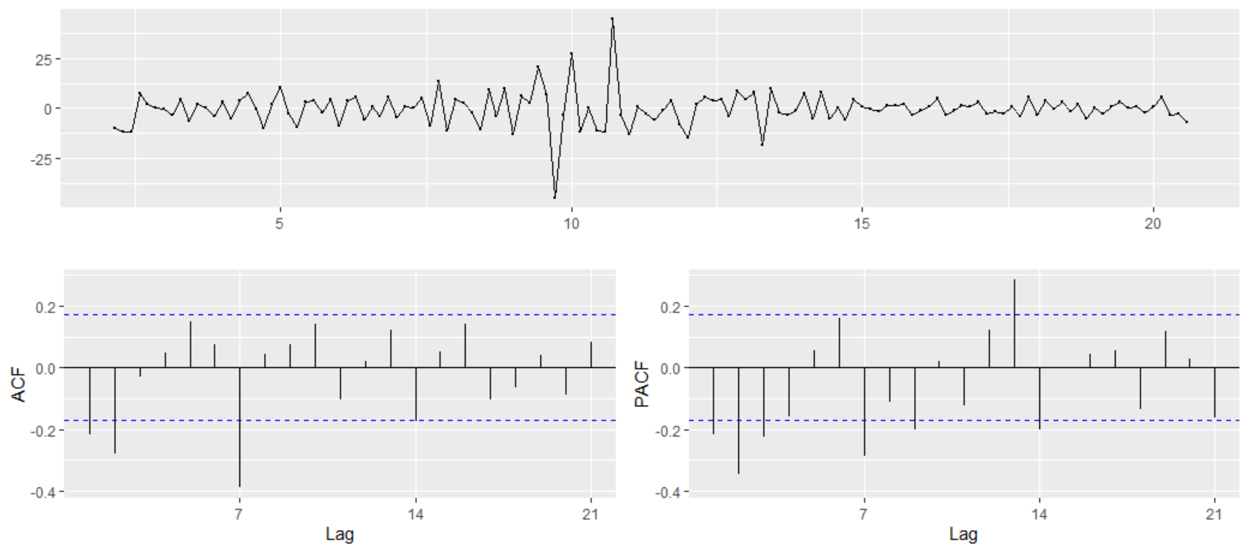

Figure 32: Time sequence plot, ACF and PACF of 2020 storage 1 after one time finite and one time seasonal differencing

In Figure 32, the time sequence plot, autoregressive function and partial autoregressive function are shown for the time series after one time finite differencing and one time seasonal differencing. In the autocorrelation function, lags 1 and 2 are above the significance level, and the first subsequent lags are not significant. This indicates that a suitable value for parameter q is 2, as seen in Section 3.5.2. Lag 7 is also above the significance level, but the seasonality period is equal to 7, so this will be covered when looking at the seasonality.
In the partial autocorrelation function, lags 1 and 2 are above the significance level, and the first lags after lag 2, a decaying behaviour can be seen. This indicates that 2 is a suitable value for parameter p, as seen in Section 3.5.1. Lag 13 is also above the significance level, but this can not be contained in the model in this part, and it might be caused by a combination of negative autocorrelations of lag 1 and 14, so this will not be taken into account at this point.
To estimate the values of parameters P and Q, the autocorrelation and partial autocorrelation functions of lag 7 and multiples of 7 can be used. For parameter Q, the autocorrelation function can be analysed. It can be seen that the contribution for lag 7 is significant, and the contribution for lag 14 is not significant. Therefore 1 is a suitable value for parameter Q. In the partial autocorrelation function it can be seen that lags 7 and 14 are significant, and lag 21 is not. Therefore 2 is a suitable value for parameter P.
All together, a fitting model can be SARIMA$(2, 1, 2)(2, 1, 1)_7$. This model is implemented in R using the *Arima* function from the *fpp2* package [2], and the results are given below.

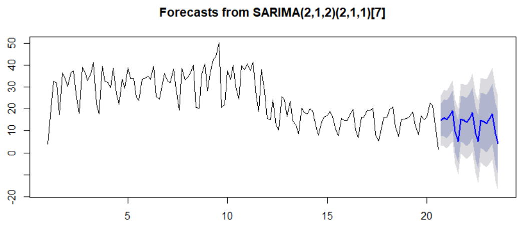

Figure 33: $\text{SARIMA}(2,1,2)(2,1,1)_7$ forecast 2020 storage 1

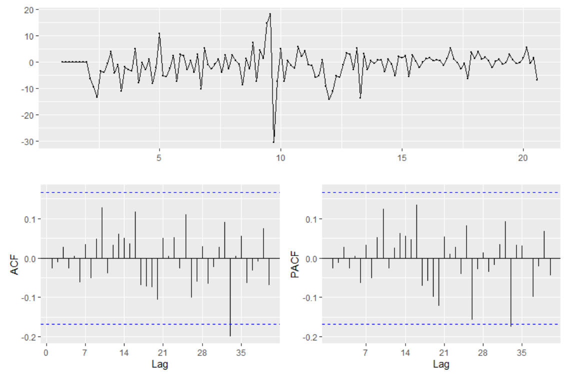

Figure 34: $\text{SARIMA}(2,1,2)(2,1,1)_7$ residuals 2020 storage 1

In Figure 33 the forecast is visualized (the blue line) with a 95% and 80% prediction interval, while in Figure 34 the in-sample residuals are plotted, together with the autocorrelation function and the partial autocorrelation function. It can be seen that there are only small autocorrelations left, which is an indication that the model fits the data well. Another indication of the quality of the model is Akaike's Information Criterion (AIC), which equals 849.66 for this model. The lower the AIC value, the better the model fits the data, where the model complexity is also taken into account. To check if this is the best model, some variations can be applied and the residuals can be studied and the AIC values can be compared to find the best fitting model.
Similar models can be applied to the data and the AIC values can be found in the table below.

Model	AIC value
$\text{SARIMA}(2,1,2)(2,1,1)_7$	849.66
$\text{SARIMA}(2,1,1)(2,1,1)_7$	847.73
$\text{SARIMA}(2,1,1)(0,1,1)_7$	844.65

The forecasts looked quite similar, and from the AIC values it can be concluded that the $\text{SARIMA}(2,1,1)(0,1,1)_7$ model fits the data best. For the $\text{SARIMA}(2,1,1)(0,1,1)_7$ model we can look at the residuals. The Ljung-Box test has a p-value of 0.9025, which means that there are no significant autocorrelation contributions left. The Shapiro-Wilk test has a p-value of 1.256e-08, which means that the assumption that the residuals are normally distributed has to be rejected.
A similar analysis is done for storage 2 and 4, which can be found in Appendix A.3.
All together, for storage 1 in 2020 the $\text{SARIMA}(2,1,1)(0,1,1)_7$ model fits the data best, for storage 2 in 2019 the $\text{SARIMA}(1,1,1)(2,1,2)_7$ model fits the data best and for storage 4 in 2019, the $\text{SARIMA}(2,1,1)(2,1,1)_7$ model fits the data best. Combining this, the $\text{SARIMA}(2,1,1)(2,1,1)_7$ is chosen to model the data, where the coefficients of polynomials $\phi_p(\boldsymbol{B})$, $\Phi_P(\boldsymbol{B})$, $\theta_q(\boldsymbol{B})$ and $\Theta_Q(\boldsymbol{B})$ will be estimated during the implementation of the model.

4.3.1 ARMA method with cut-off at zero

In Figures 35 and 36, the ARMA forecasts for storage 1 and 4 are visualised where the forecasts are cut-off below zero, which means that all negative forecasts are set to zero.

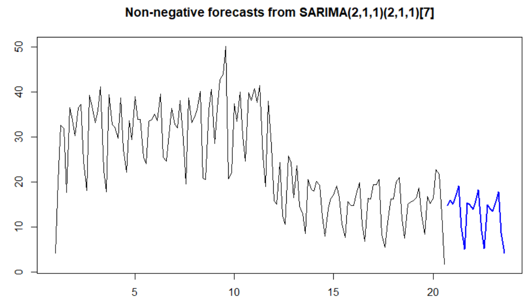

Figure 35: ARMA forecast with cut-off at zero for 2020 storage 1

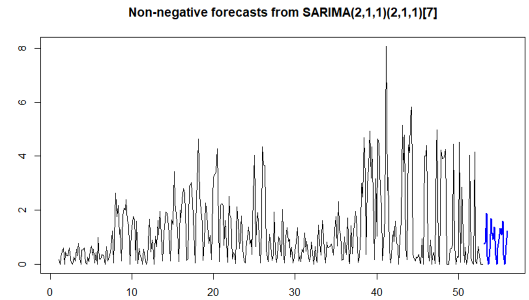

Figure 36: ARMA forecast with cut-off at zero for 2019 storage 4

It can be seen that the forecasts are the same as for the classic ARMA method, but the values below zero are all set to zero. An advantage of this method is that the other forecasts remain the same, especially for storages that do not have sales close to zero, nothing changes in the forecasts. A disadvantage is that the forecasts have to be corrected, which means that the patterns and constraints are not fully captured by the model.

4.3.2 ARMA method with Box-Cox transformation

Another way to produce forecasts that are non-negative is by using a Box-Cox transformation in combination with the ARMA method. As found in Section 4.3, the SARIMA$(2,1,1)(2,1,1)_7$ model is chosen. For the Box-Cox transformation $\lambda_1 = 0$ is chosen and λ_2 is half of the smallest non-zero value in the data.
For the application of the Box-Cox transformation in combination with the ARMA forecasting method, first the observations will be transformed using the Box-Cox transformation. Next the transformed values will be used to construct forecasts, and lastly the forecasts will be transformed back using the inverse of the Box-Cox transformation.

First we will look at the results for storage 1 in 2020.

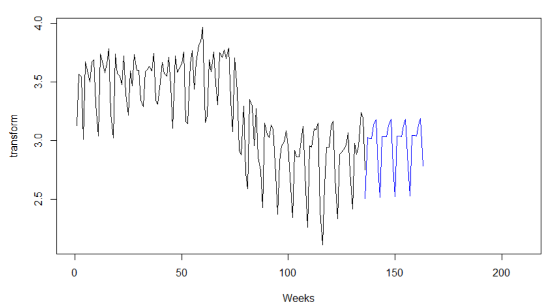

Figure 37: ARMA forecast with Box-Cox transformation for 2020 storage 1, transformed scale

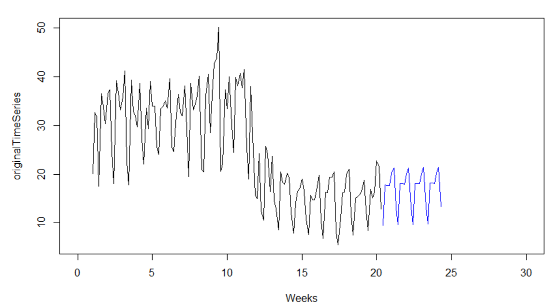

Figure 38: ARMA forecast with Box-Cox transformation for 2020 storage 1, original scale

In Figure 37 the Holt-Winters forecast with bootstrapping can be seen in the Box-Cox transformed scale. In the forecast, the seasonality pattern can be seen, and there seems to be a small trend in the forecast. In Figure 38 the forecast in the original scale can be seen, so this is the forecast after the inverse Box-Cox transformation is applied. It can be seen that the forecasts on the transformed and original look quite similar. In Section 4.6 the forecasts will be compared to observations.

A similar analysis is done for 4 in 2019, which contains zero values.

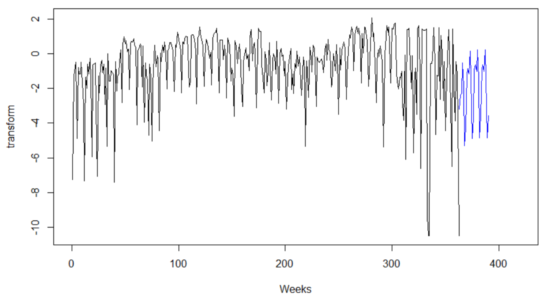

Figure 39: ARMA forecast with Box-Cox transformation for 2019 storage 4, transformed scale

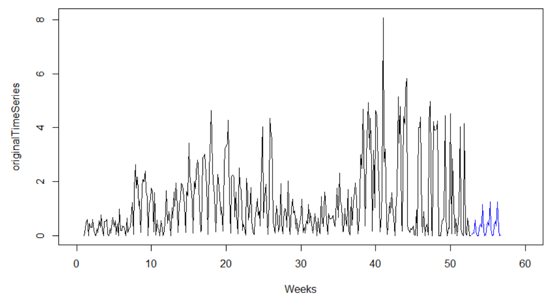

Figure 40: ARMA forecast with Box-Cox transformation for 2019 storage 4, original scale

In Figure 39 the Holt-Winters forecast with bootstrapping can be seen in the Box-Cox transformed scale. In the forecast, the seasonality pattern can be seen, and there seems to be a small trend in the forecast. On sight the forecasts seem to be quite low compared to the historic observations. In Figure 40 the forecast in the original scale can be seen. It can be seen that there are no negative forecasts, and the pattern of the zero values seems to be accurate. However, the forecasts seem to be very low compared to the historic observations.

All together, a forecast that is based on a Box-Cox transformation combined with the Holt-Winters model with bootstrapping produces non-negative forecasts. However the forecasts overall seem to be less accurate, and this will be investigated more in Section 4.6 where the out-of-sample evaluation will be performed.

4.3.3 ARMA method using bootstrapping and Box-Cox transformation

Another way to produce forecasts that are non-negative is by using a Box-Cox transformation in combination with the ARMA method and bootstrapping. As found in Section 4.3, the SARIMA$(2,1,1)(2,1,1)_7$ model is chosen. Again for the Box-Cox transformation $\lambda_1 = 0$ is chosen and λ_2 is half of the smallest non-zero value in the data.
The procedure is performed in R, where the moving block bootstrap is performed using the *bld.mbb.bootstrap* function from the $forecast$ package [2].

First we will look at the results for storage 1 in 2020.

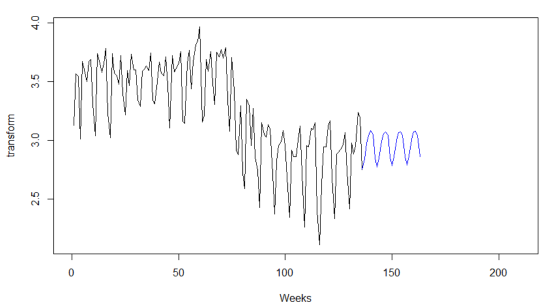

Figure 41: ARMA forecast with bootstrapping for 2020 storage 1, transformed scale

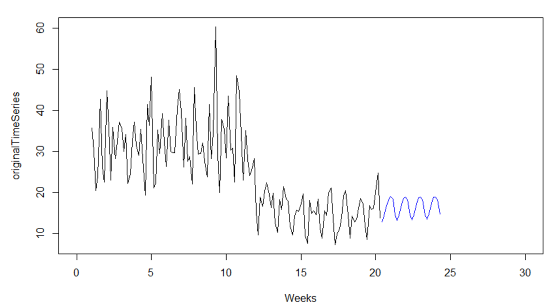

Figure 42: ARMA forecast with bootstrapping for 2020 storage 1, original scale

In Figure 41 the ARMA forecast with bootstrapping can be seen in the Box-Cox transformed scale. It can be seen that the amplitude of the forecast is smaller than the amplitude of the observations. It can also be seen that there is a seasonal pattern, but it is more smooth than the seasonal pattern in the observations.

In Figure 42 the forecast in the original scale can be seen, so this is the forecast after the inverse Box-Cox transformation is applied. It can be seen that the forecasts on the transformed and original scale look quite similar, and again the forecasts are more smooth and have a smaller amplitude than the historic observations. In Section 4.6 the forecasts will be compared to observations.

A similar analysis is done for 4 in 2019, for which the sales contain zero values.

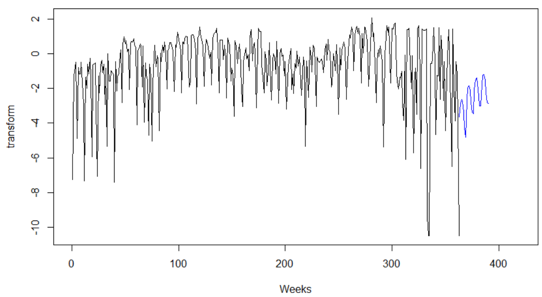

Figure 43: ARMA forecast with bootstrapping for 2019 storage 4, transformed scale

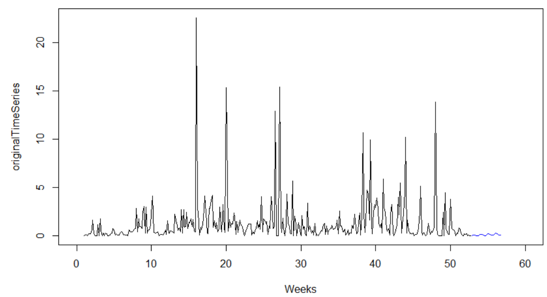

Figure 44: ARMA forecast with bootstrapping for 2019 storage 4, original scale

In Figure 43 the ARMA forecast with bootstrapping can be seen in the Box-Cox transformed scale. In the forecasts the seasonality pattern can be seen together with and an upward trend.
In Figure 44 the forecast in the original scale can be seen. There are no negative forecasts, and the pattern of the zero values seems to be accurate. However, the forecasts seem to be very low compared to the historic observations.

All together, a forecast that is based on a Box-Cox transformation combined with the ARMA model with bootstrapping produces non-negative forecasts. However the forecasts overall seem to be less accurate, and this will be investigated more in Section 4.6 where the out-of-sample evaluation will be performed.

4.4 Combined method

Bottomline has a lot of storages for which forecasts have to be calculated, and because of the large amount of storages, also a lot of differences can be found when comparing storages. This causes that one method might work best for some storages, and another method works best for other storages. An option to deal with this is to determine for every storage which method can be used best. This can be done by fitting all methods for the historic data used except for the last week, and calculating forecasts for last week. Then the MSE can be calculated for every method, and the method with the smallest MSE is chosen to calculate forecasts for the week of interest.
For storages 1, 2 and 4, the MSE of the last week is shown (the week on which the method for next week is based), and the MSE of the next week is shown.

	Storage 1, week before	Storage 1, this week	Storage 2, week before	Storage 2, this week	Storage 4, week before	Storage 4, this week
MSE Holt-Winters	8,034465	3,966552	1,415342	0,177798	2,987349	3,220679
MSE Box-Cox HW	6,010972	1,67517	1,444577	0,137645	3,685335	3,93565
MSE Bootstrapped HW	9,999969	2,073703	1,339986	0,271184	4,186429	3,974807
MSE ARMA	4,72521	1,800537	0,22734	0,257226	2,158034	2,332628
MSE Box-Cox ARMA	3,822974	0,689396	0,596127	0,218291	2,749507	3,586418
MSE Bootstrapped ARMA	5,459242	0,96126	0,636508	0,296118	3,747438	3,753226
MSE Current method	123,5531	117,2845	0,117009	0,377935	1,96279	2,572761
MSE Combination method		0,689396		0,377935		2,572761

For storage 1 it can be seen that on the week before, the ARMA method with Box-Cox transformation was chosen. In the results of the next week it can be seen that the ARMA method with Box-Cox transformation is the best method, so for this storage the combination method gives the forecasts with the best MSE.
For storage 2 it can be seen that based on the week before, the current Bottomline method is chosen. For the next week the Holt-Winters method with a Box-Cox transformation is chosen. This time, the classic Bottomline method gives the worst MSE, and therefore also the combination method gives the worst MSE.
For storage 4 it can be seen that based on the week before, the current Bottomline method is chosen. For the next week the ARMA method is chosen, but the MSE of the method that was chosen based on the week before is close to the MSE of the best method.
All together, the idea of choosing the best method based on one week before works well for some storages, but unfortunately it does not give the best forecasts for all storages. A conclusion from this is that the correlation between which method is better for subsequent weeks is not very high, and therefore it is not sure yet if this will produce the best forecasts. In Section 4.6 a simulation is used to do a out-of-sample evaluation for a lot of time series which can be used to see more reliable results.

4.5 Amount of data per method

Some methods are more complex than other methods and in general more complex methods need more data to calculate good predictions. A study is performed to find out what the best amount of data is for the different methods to make the best predictions. First some literature research is done, and second the data of Bottomline is used to determine the best amount of data for every method.
One way to determine the best amount of data per method is by using literature. To do this, the books and articles used in Chapter 3 are investigated. It is studied what amount of data is used in the exampled used in literature in the chapters corresponding to the different methods. It is found that for both Holt-Winters and SARIMA forecasting methods, the amount of data used in literature varies between 10 and 20 times the seasonality period.
The amount of data needed to forecast depends on the type of product and the noise in the data. For example when a product is revised a lot, it is wise to use a short period of data. In this project, the product is fuel, which is a relatively stable product and therefore a longer period of data can be used.
Another aspect is the noise in the data. When there is a lot of noise in the data, it is harder to capture the underlying patterns and more data is needed to do so. When there is not much noise in the data, a shorter period of data can be used to produce forecasts with a small MSE.
To determine the amount of data to use per method, all described methods are fit multiple times on different periods of data with lengths varying from 5 to 40 weeks, and this is done for multiple storages. For every storage and amount of data, a moving time window is used to fit all methods multiple times using the same amount of data (but a different period) for one storage, which gives a procedure similar to cross-validation. Every time each model is fit on the given data, the forecasts for one week are calculated and the Mean Squared Error is calculated. Per storage for the same amount of data, the MSE values are averaged, and then per storage the average MSE per method is compared for all lengths of data. This way it can be seen per storage and per method what the best amount of data is to calculate the most accurate forecasts.
It is seen that the amount of data for the best forecast per method differs between storages. This can be explained by the fact that different storages have different levels of noise in the data. It is also seen that per storage, the best amount of data is not very different for different methods. This is remarkable because the models do not all have the same complexity.
The best amount of data varies for all methods from 10 to 40 weeks, but averaging the best amount of data for all storages gives that a period of 30 weeks is best to use for all methods. It can be seen that the amount of data depends more on the storage and the noise in the data, than it depends on the forecasting method that is used.

4.6 Compare all results

To be able to evaluate and compare the out-of-sample fitting of all implemented methods, a lot of time series are needed. Also it is important to make sure that the results are generally applicable, so not only for the

few storages that are used for the evaluation. This can be obtained by generating time series, where the parameters can be based on real storages to make sure that the generated time series are realistic and a good representation of a real storage.
An advantage of using a simulation to construct and fit data is that a lot more data can be used and therefore the results are more significant and general. Also, when the time series are generated, the "correct parameters" for trend and seasonality are known and the effect of random noise and simulated disturbances can be evaluated more accurate.
To construct time series that are representative for real storages, a couple of storages are analysed to estimate parameters for the simulations. The time series are constructed using a trend component, a seasonality component and a random component. The trend has a mean which is the starting point, and from there every time step, a sample of a normally distributed random variable with mean zero is added. This way a random walk is obtained for the trend. Lastly disturbances are added randomly, on average once every month. With probability of 0.5 the an observation is changed in a zero, and with probability 0.5 an observation is multiplied with a disturbance factor. This disturbance factor is sampled from a Normal distribution with mean 1 and variance 0.4, which is based on knowledge data.
The simulation is used to generate 1000 time series using the parameters of 11 real storages, so this way 11000 time series are simulated. One of the simulated time series is visualised in the following time series decompositions.

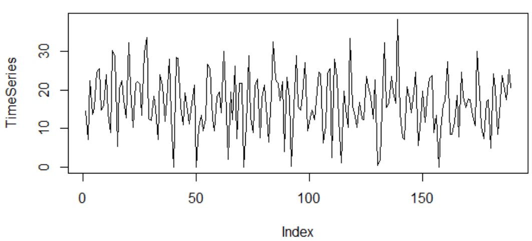

Figure 45: Generated time series

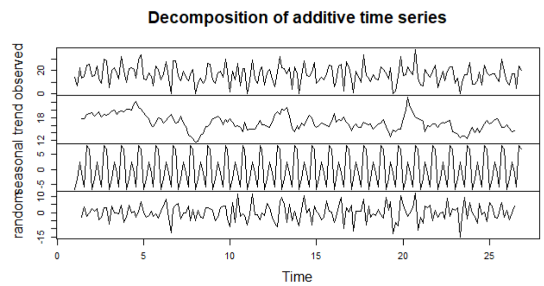

Figure 46: Time series decomposition of generated time series

When this series is analysed, approximately the same parameters are found as when analysing a real storage. Therefore the time series that are constructed are representative for reality and can be used to evaluate the forecasting methods.

For each of the simulated time series, one week of forecasts is constructed, and the total mean squared error is calculated. This is done for 30 weeks of data, because in Section 4.5 was found that this is the best amount of data for all methods. In the simulation of the time series, disturbances are added on average once every two months. All investigated methods are implemented, and also the MSE_perfect is calculated, which is theoretically the perfect score, because this is the MSE of the observations compared to the generated time series without random term and disturbances. The results of the simulation are given in the table below.

MSE Holt-Winters method	10,27503
MSE Holt-Winters method with Box-Cox transformation	12,47887
MSE Holt-Winters method with Box-Cox transformation and bootstrapping	12,47857
MSE Bottomline method	10,19104
MSE ARMA method	9,429592
MSE ARMA method with Box-Cox transformation	10,89318
MSE ARMA method with Box-Cox transformation and bootstrapping	10,95175
MSE Combination method	9,599953
MSE perfect	8,856616

Table 1: Results of all methods

In the table it can be seen that the ARMA method gives the best results, and the combination method gives the second best results. The methods with Box-Cox transformation or Box-Cox transformation and bootstrapping give the worst results. This can be caused by the fact that those methods also change the forecasts when the sales are not close to zero.
It can be observed that the combination method does give good results, but is not the best of all methods. This means that the correlation between which method had the lowest MSE for the forecasts last week, and which method has the lowest MSE for the forecasts this week is not very high.

To visualize the results, historic data is shown and forecasts are calculated for all different methods and shown in different colours in the figure below.

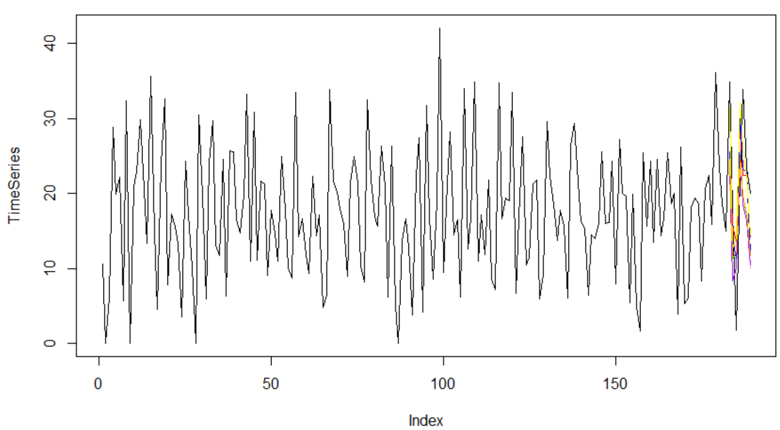

Figure 47: Forecasts for generated time series

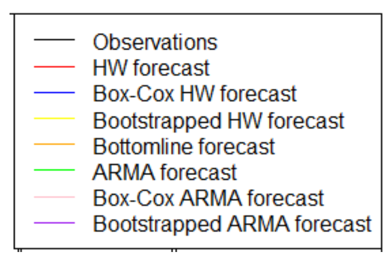

Figure 48: Legend

It can be seen that the forecasts of all methods are close to each other.

Another interesting aspect to evaluate are the residuals of the forecasts compared to the observations. The histograms of the residuals are shown below for the different methods.

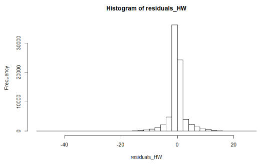

Figure 49: Residuals of Holt-Winters forecasts

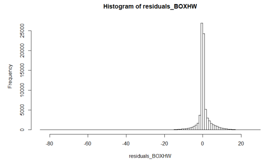

Figure 50: Residuals of Holt-Winters forecasts with Box-Cox transformation

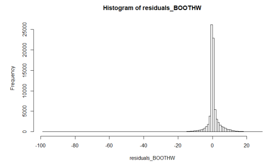

Figure 51: Residuals of Holt-Winters forecasts with Box-Cox transformation and bootstrapping

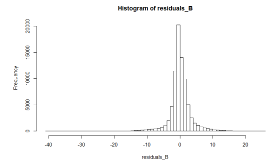

Figure 52: Residuals of Bottomline forecasts

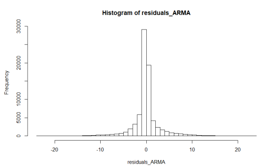

Figure 53: Residuals of ARMA forecasts

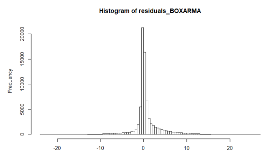

Figure 54: Residuals of ARMA forecasts with Box-Cox transformation

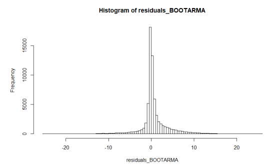

Figure 55: Residuals of ARMA forecasts with Box-Cox transformation and bootstrapping

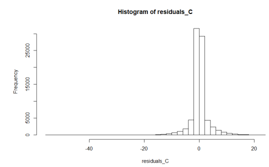

Figure 56: Residuals of combined method forecasts

It can be seen that the residuals of all methods are centered symmetrically around zero, and no big differences are found.
All together it can be concluded that the ARMA method calculates the best forecasts, where the $\text{SARIMA}(2,1,1)(2,1,1)_7$ model is used.

5 Conclusion and advice

The goal of this research was to investigate if the inventory forecasts can be improved to become more accurate. To do this, data of Bottomline was explored and new models were applied to see if the forecasts were more accurate.
In the analysis of the data and methods it has been seen that there are a lot of storages that have differences and similarities. Most storages have a weekly pattern, but the average sales and weekly patterns differ a lot. Also the trend component differs a lot between storages.
It has been seen that overall the ARMA method performs best, and both the ARMA method and the combination method give better results than the currently used Bottomline method.
Therefore, the advice to Bottomline is to use the ARMA method in the future to predict the sales forecasts. The ARMA method calculates the best forecasts when 30 weeks of data are used. Therefore the advice for Bottomline is to increase the amount of data used to calculate forecasts to 30 weeks. In fact, also the Bottomline method produces best forecasts when on average 30 weeks of data are used, to this can also be an improvement compared to the forecasting method that is currently used.

6 Discussion and further research

It is concluded that the ARMA method is the best method to calculate forecasts for the inventory. However, there are some limitations to the research that will be discussed in this section.

First, there are a lot of storages and there are a lot of differences between storages. In this research only storages were considered of which daily data was available to make sure the results do not depend on the implemented weekly pattern, but on real data. However, there are also a lot of storages that do not have daily data and the sales per day are based on the weekly pattern. To make sure those sales and forecasts are also as accurate as possible, it can be useful to check if the weekly patterns are up to date, or maybe if the weekly patterns can be improved.
Also the knowledge is an interesting part of the process where maybe progress can be made. Currently the owner of a gas station has to set when a ”special day” will occur, and for that day what the multiplication factor of the sales will be compared to the normal situation. This can be a simple task for example if the station is closed on a holiday, but in other cases it can be a very difficult task for the station owner. To improve this, historic data of the special day or event can be used to help estimate the multiplication factor, and maybe this can be done automatically.
An improvement that might be made to the forecasting methods is to use both the past weeks, and the information about the same day in the past years. However this is difficult because the same date a year earlier is another day of the week and some holidays are not always on the same date, etc.

References

[1] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis : forecasting and control.* Hoboken, New Jersey: John Wiley & Sons, Inc., fifth edition / ed., 2016.

[2] R. J. Hyndman and G. Athanasopoulos, *Forecasting : principles and practice.* Heathmont, Vic.: OTexts, 2014.

[3] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to time series analysis and forecasting.* Hoboken, New Jersey: Wiley, second edition. ed., 2015.

[4] S. P. Millard, “Envstats : an r package for environmental statistics,” 2013.

[5] C. Bergmeir, R. J. Hyndman, and J. M. Benitez, “Bagging exponential smoothing methods using stl decomposition and box-cox transformation,” *International Journal of Forecasting*, vol. 32, no. 2, p. 303, 2016. 303.

[6] R. M. Sakia, “The box-cox transformation technique: A review,” *Journal of the Royal Statistical Society. Series D (The Statistician)*, vol. 41, no. 2, pp. 169–178, 1992. 169.

[7] R. J. Hyndman, “Transforming data with zeros.” https://robjhyndman.com/hyndsight/transformations/, 2010.

[8] J. M. G. Taylor, “The retransformed mean after a fitted power transformation,” *Journal of the American Statistical Association*, vol. 81, no. 393, pp. 114–118, 1986. 114.

[9] E. M. de Oliveira and F. L. Cyrino Oliveira, “Forecasting mid-long term electric energy consumption through bagging arima and exponential smoothing methods,” *Energy*, vol. 144, pp. 776–788, 2018. 776.

A Appendices

A.1 Exploratory data analysis

In addition to the exploratory data analysis in Section 4.1, the sales of storage 2 in 2019 will be analysed. The time series decomposition is shown below.

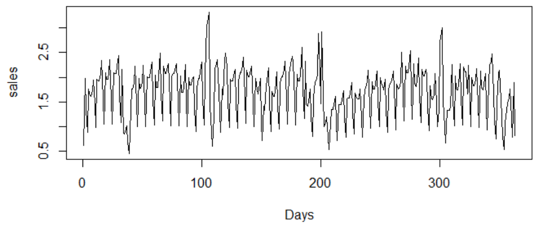

Figure 57: Time sequence plot 2019 storage 2

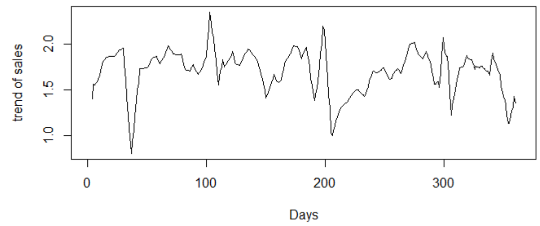

Figure 58: Trend 2019 storage 2

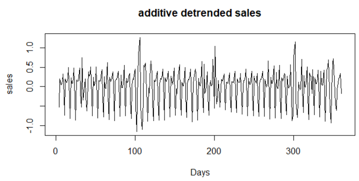

Figure 59: Series without trend 2019 storage 2

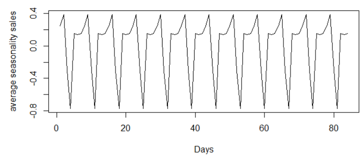

Figure 60: Seasonality 2019 storage 2

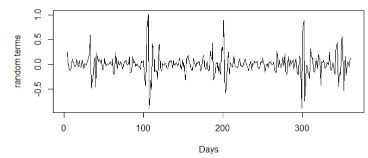

Figure 61: Residuals 2019 storage 2

Just like for storage 1, it can be seen that there is a clear seasonal pattern in the series after the trend is removed.
The trend is very inconstant which can be caused by the period length of the moving average function, which is 7 days in this case. For example if a period of 28 days is used, the trend is much more smooth, as can be seen in Figure 62.

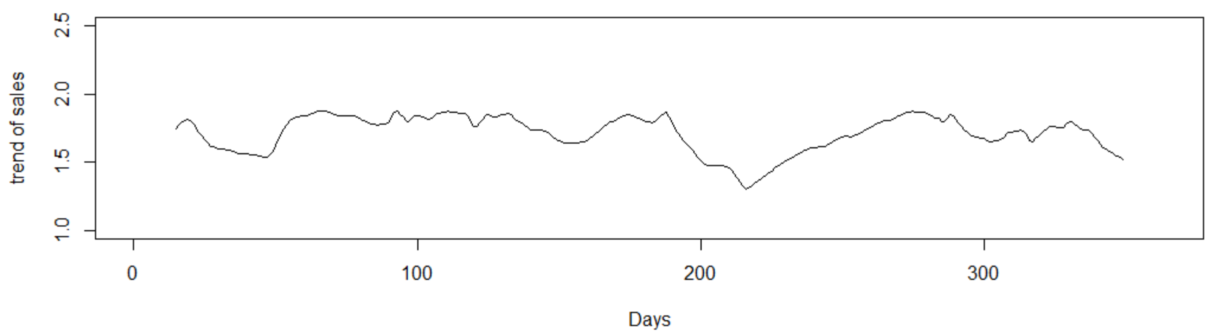

Figure 62: Trend of storage 2 in 2019 with moving average window of 28 days

The mean level of the sales is not constant, but also no very clear trend can be seen. The fluctuations might be caused by some seasonal pattern or they can be caused by a trend factor. The presence of trend can be evaluated in more detail in the finite differencing analysis.
The time sequence plot (also shown in Figure 57), autocorrelation function plot and partial autocorrelation plot of storage 2 in 2019 are shown in Figure 63.

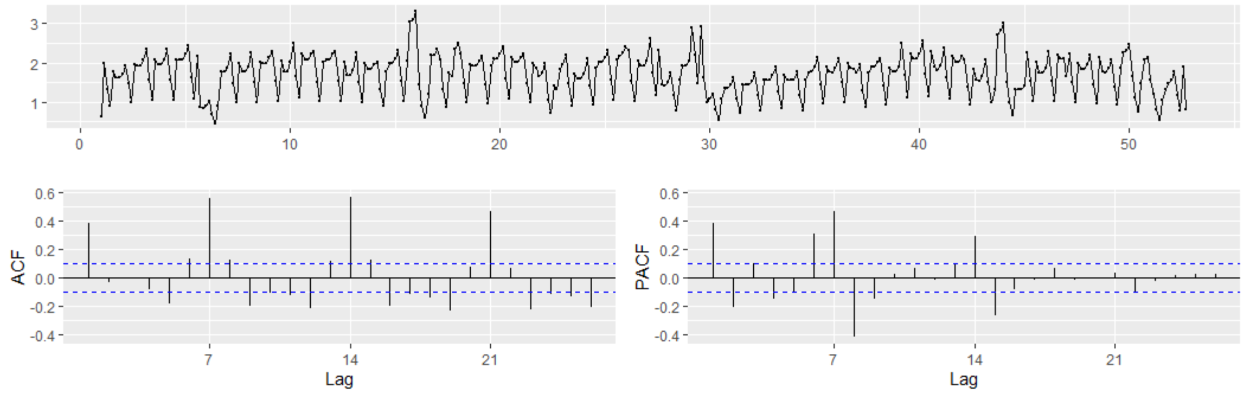

Figure 63: Exploratory data analysis of storage 2 in 2019

It can be seen that the autocorrelations at lag 1, and lags multiples of 7 are significant, where the significance at lag 1 can imply trend and lags multiples of 7 can imply seasonality. To remove the significant autocorrelation at lag 1, finite differencing can be applied. The obtained time series and autocorrelation functions are shown below.

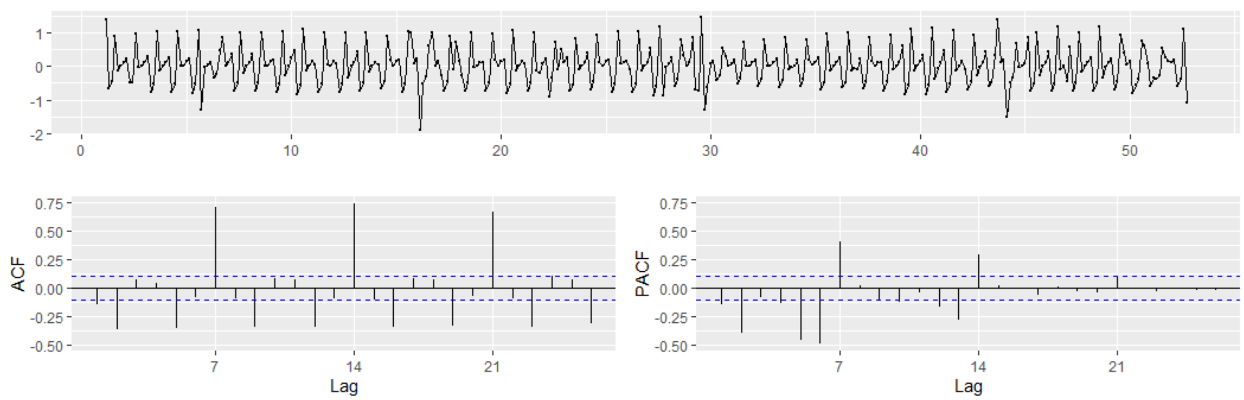

Figure 64: One time finite differencing storage 2 in 2019

It can be seen that the significant autocorrelation contribution at lag 1 is removed after one time finite differencing, so this is useful to do. There are still big contributions at lags multiples of 7, so in order to obtain a stationary time series, also seasonal differencing is applied with a seasonality period of 7 days. The result is shown in Figure 65.

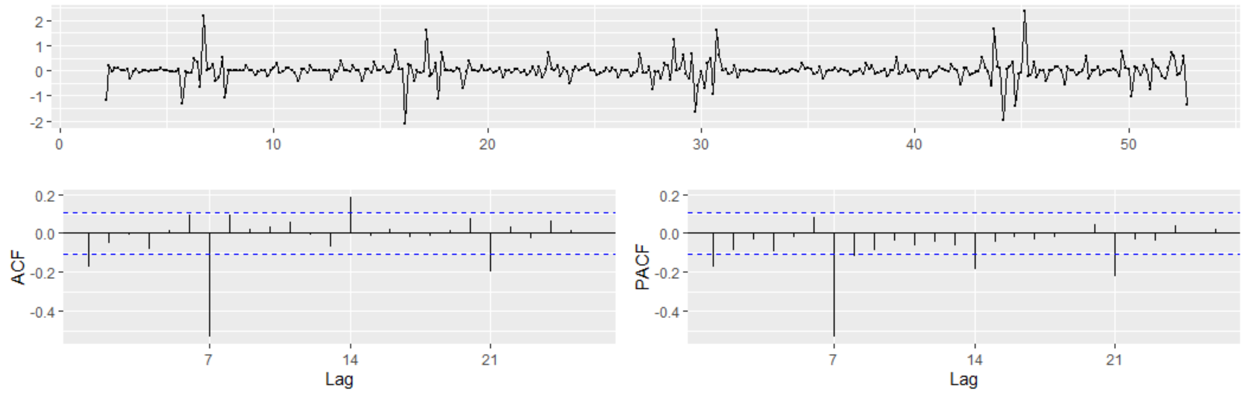

Figure 65: One time finite differencing and one time seasonal differencing storage 2 in 2019

The contributions of lags multiples of 7 to the autocorrelation function and partial autocorrelation are still significant, but they are smaller than before the seasonal differencing. An option can be to apply seasonal differencing a second time, and the results are shown in below.

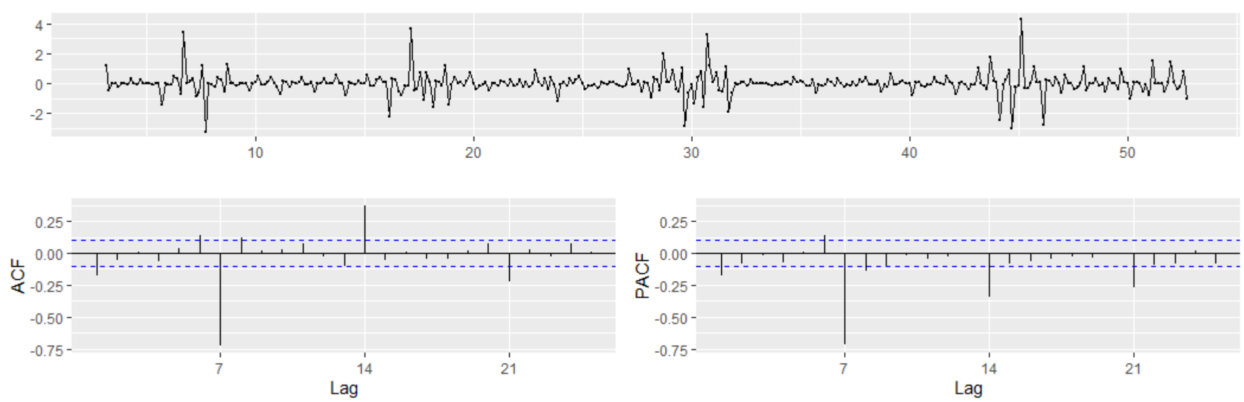

Figure 66: One time finite differencing and two times seasonal differencing storage 2 in 2019

In Figure 66 it can be seen that after a second time applying seasonal differencing, the autocorrelation contributions become higher, which is not desirable. Therefore the best option is applying one time finite differencing and one time seasonal differencing with a seasonality period of 7 days.
To be able to base decisions on multiple storages, the sales of storage 3, 4 and 5 in 2019 will be analysed, where storage 3 sells a lot compared to storage 4 and 5. The time series decompositions are shown below, where the decomposition steps are visualised for all three storages at the same time.

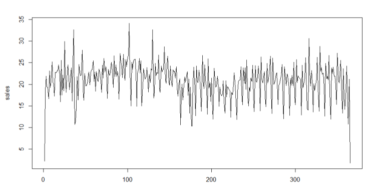

Figure 67: Time sequence plot 2019 storage 3

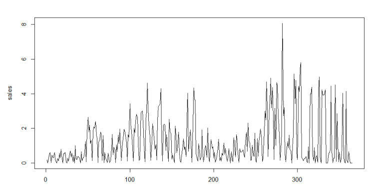

Figure 68: Time sequence plot 2019 storage 4

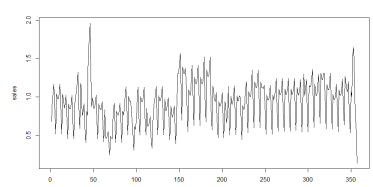

Figure 69: Time sequence plot 2019 storage 5

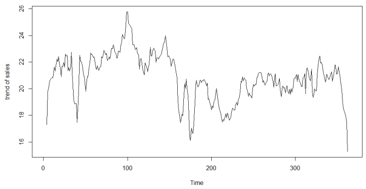

Figure 70: Trend with moving average window of 7 days 2019 storage 3

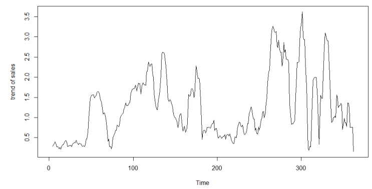

Figure 71: Trend with moving average window of 7 days 2019 storage 4

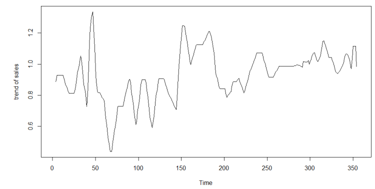

Figure 72: Trend with moving average window of 7 days 2019 storage 5

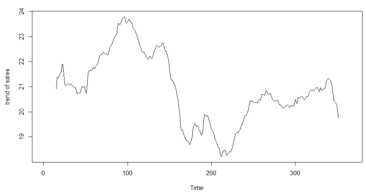

Figure 73: Trend with moving average window of 28 days 2019 storage 3

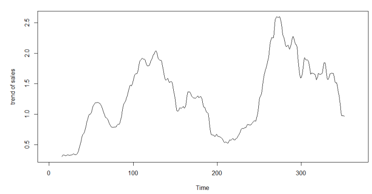

Figure 74: Trend with moving average window of 28 days 2019 storage 4

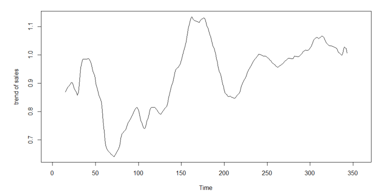

Figure 75: Trend with moving average window of 28 days 2019 storage 5

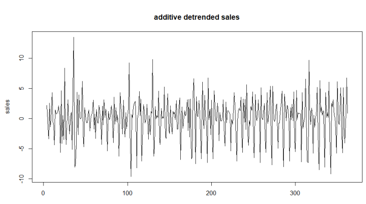

Figure 76: Detrended time series 2019 storage 3

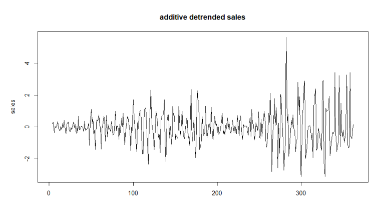

Figure 77: Detrended time series 2019 storage 4

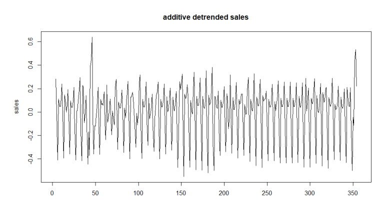

Figure 78: Detrended time series 2019 storage 5

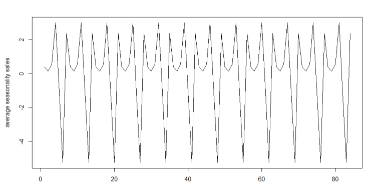

Figure 79: Average seasonality 2019 storage 3

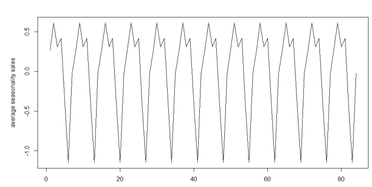

Figure 80: Average seasonality 2019 storage 4

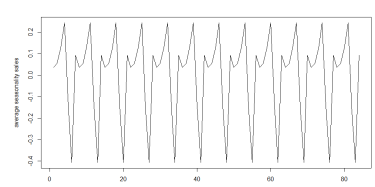

Figure 81: Average seasonality 2019 storage 5

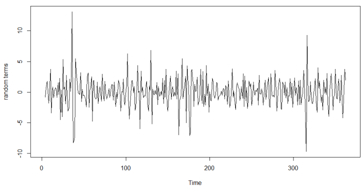

Figure 82: Random terms 2019 storage 3

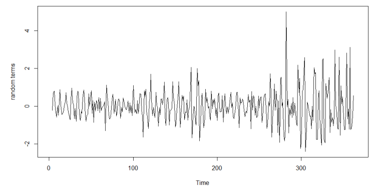

Figure 83: Random terms 2019 storage 4

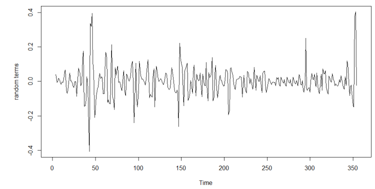

Figure 84: Random terms 2019 storage 5

In the time series decompositions above it can be seen that for both storage 3, 4 and 5 there is a trend component which is not constant over time (both for the moving average window of 7 days and 28 days). For all three storages it can also be seen that there is a clear seasonal pattern.

The autocorrelation functions, partial autocorrelation functions and differencing will be analysed below.

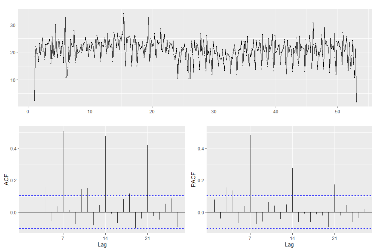

Figure 85: Exploratory data analysis 2019 storage 3

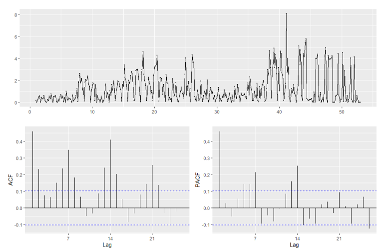

Figure 86: Exploratory data analysis 2019 storage 4

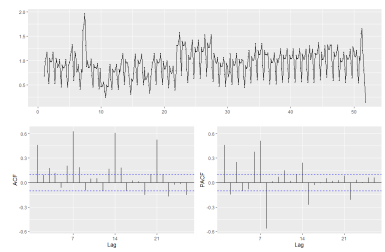

Figure 87: Exploratory data analysis 2019 storage 5

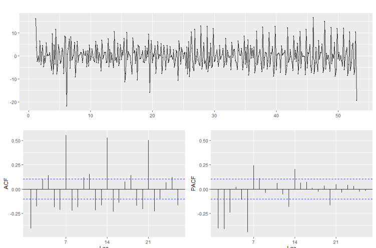

Figure 88: One time finite differencing 2019 storage 3

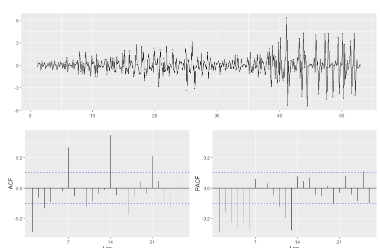

Figure 89: One time finite differencing 2019 storage 4

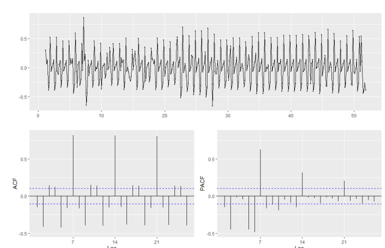

Figure 90: One time finite differencing 2019 storage 5

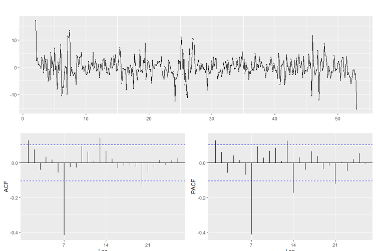

Figure 91: One time seasonal differencing 2019 storage 3

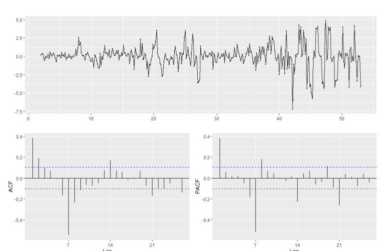

Figure 92: One time seasonal differencing 2019 storage 4

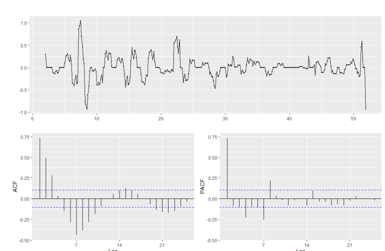

Figure 93: One time seasonal differencing 2019 storage 5

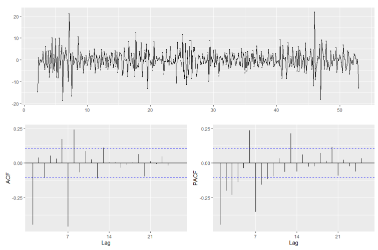

Figure 94: One time finite differencing and one time seasonal differencing 2019 storage 3

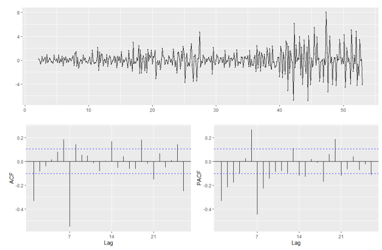

Figure 95: One time finite differencing and one time seasonal differencing 2019 storage 4

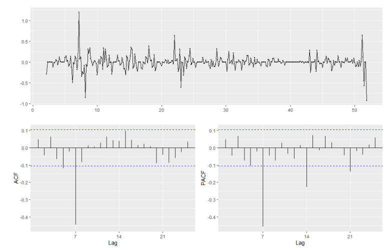

Figure 96: One time finite differencing and one time seasonal differencing 2019 storage 5

In the figure above it can be seen that for all three storages, at least one time seasonal differencing is needed to remove the seasonal pattern. For both storage 4 and 5, also one time finite differencing is needed to remove the trend. For storage 3 this is less needed.

A.2 Exponential smoothing

In addition to the exponential smoothing analysis in Section 4.2, the Holt-Winters method is applied for storage 2 in 2019. The obtained smoothing parameter estimates are the following.

$$\begin{aligned}\hat{\alpha} &= 0.7908583\\ \hat{\beta} &= 0.005735719\\ \hat{\gamma} &= 0.114495\end{aligned}$$

The smoothing parameter $\hat{\alpha}$ is close to one, which means that with respect to the level estimate, more weight is given to recent observations, compared to observations further in the past. The values of $\hat{\beta}$ and $\hat{\gamma}$ are close to zero, so with respect to the trend and the seasonality, more weight is given to observations further in the past.

The values of $\hat{\beta}$ and $\hat{\gamma}$ are close to the smoothing values of storage 1, but the value of $\hat{\alpha}$ is much higher for this storage. This causes the level component to be less smooth, and allows for bigger fluctuations in the level estimates for different points in time.

The estimates for the level, trend and seasonality components for the last time step T are the following.

$$L_T = 0.8677885370$$

$$T_T = -0.0002887829$$

$$S_T = \begin{bmatrix} S_{T-s} \\ S_{T-s+1} \\ \vdots \\ S_T \end{bmatrix} = \begin{bmatrix} 0.1106913235 \\ 0.1544185872 \\ 0.2972816018 \\ -0.3292897261 \\ -0.7403597110 \\ 0.1946450177 \\ 0.1358136491 \end{bmatrix}$$

In Figures 97 and 98, the in-sample forecast can be seen, together with the Holt-Winters forecast for 3 weeks in the future.

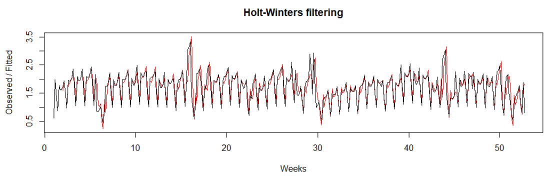

Figure 97: In-sample Holt-Winters forecast 2019 storage 2

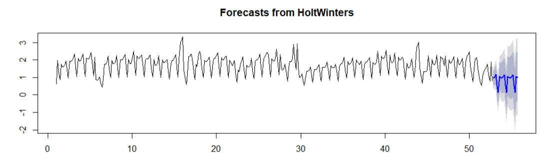

Figure 98: Holt-Winters forecast 2019 storage 2

In Figure 97 it can be seen that the in-sample forecast captures the sales very well, but this is not surprising as the same data is used for fitting and validation of the model. The forecast for the future, visualized in Figure 98, shows that the seasonal pattern is well captured, but the level seems to be a bit low compared to the sales observations. To obtain more information about the quality of the model, the forecasts will be compared to observed data in Section 4.6.

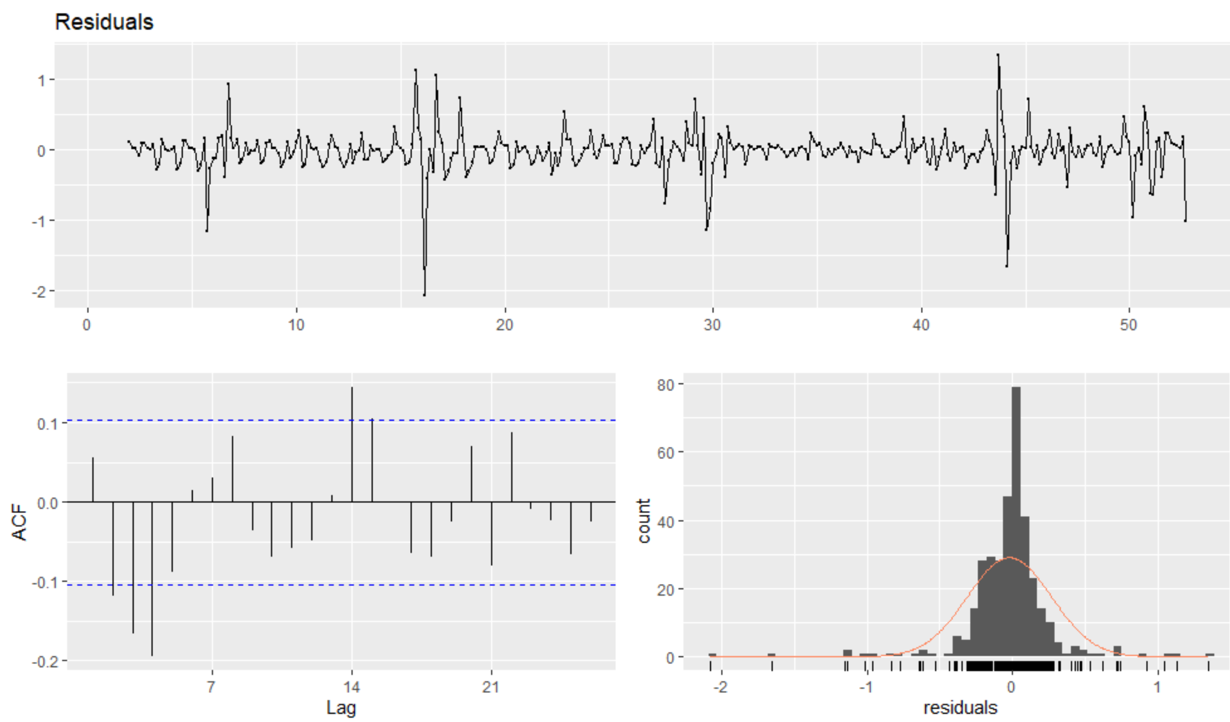

Figure 99: Residuals of in-sample Holt-Winters forecast 2019 storage 2

In Figure 99 the time sequence plot, the autocorrelation function and a histogram can be seen of the residuals of the in-sample forecast. The residuals are centered around zero, but there are still some significant autocorrelation contributions. The Ljung-Box test is performed with significance level $\alpha = 0.05$ to check if the residuals are uncorrelated. The p-value equals 0.0000383 which means that the null-hypothesis of uncorrelated residuals is rejected. This means that there is information left in the residuals that can be used to improve the forecast. In the histogram it can be seen that the residuals are centered around zero and the shape looks like a Bell-curve, but the figure is too spiked to have normally distributed residuals.

Next the Holt-Winters method is applied for storage 4 in 2019. The obtained smoothing parameter estimates are the following.

$$\hat{\alpha} = 0.2785497$$
$$\hat{\beta} = 0$$
$$\hat{\gamma} = 0.1448429$$

The smoothing parameters $\hat{\alpha}$ and $\hat{\gamma}$ are close to zero, which means that with respect to the level and the seasonality, more weight is given to observations further in the past. The smoothing parameter $\hat{\beta}$ is zero, which means that for the trend component, only the trend observation of one step earlier is used, and no observations further in the past are used.
The estimates for the level, trend and seasonality components for the last time step T are the following.

$$L_T = 0.211270771$$

$$T_T = -0.008016537$$

$$S_T = \begin{bmatrix} S_{T-s} \\ S_{T-s+1} \\ \vdots \\ S_T \end{bmatrix} = \begin{bmatrix} 0.922142697 \\ 0.470650120 \\ 1.287178542 \\ -0.631676419 \\ -1.232318541 \\ -0.157678806 \\ 0.366513310 \end{bmatrix}$$

In the figures below, the in-sample forecast can be seen, together with the Holt-Winters forecast for 3 weeks in the future.

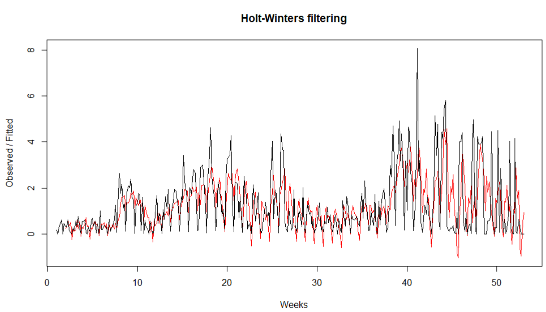

Figure 100: In-sample Holt-Winters forecast 2019 storage 4

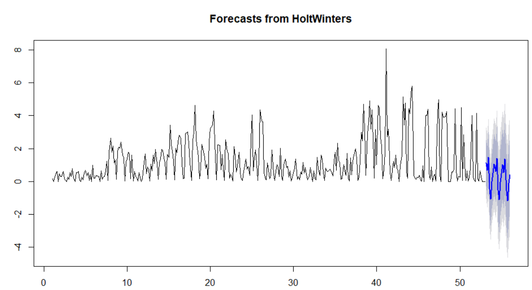

Figure 101: Holt-Winters forecast 2019 storage 4

In Figure 100 the in-sample forecast is shown. The seasonal pattern is captured well, but the forecasted values are sometimes below zero, which is not possible in practice.
The forecast for the future, visualized in Figure 101, shows that the seasonal pattern is well captured, but again there are forecasted values below zero, which is not possible in practice.
In the figure below, the time sequence plot, the autocorrelation function and a histogram can be seen of the residuals of the in-sample forecast.

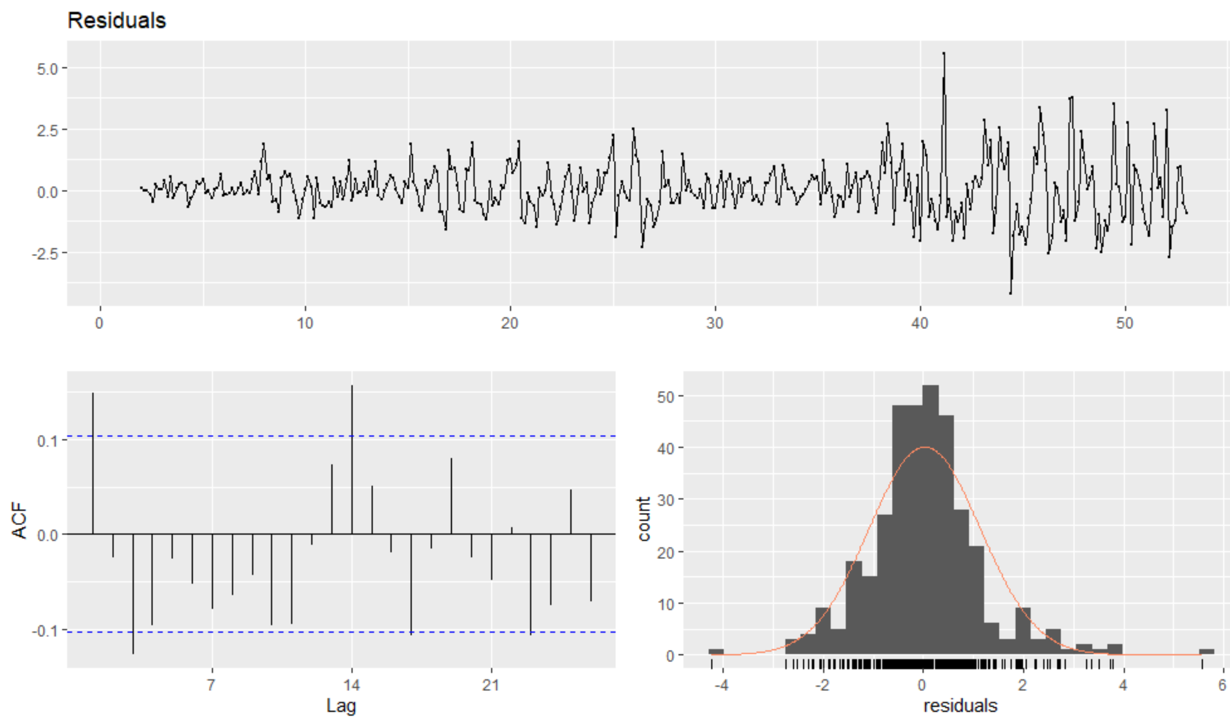

Figure 102: Residuals of in-sample Holt-Winters forecast 2019 storage 4

In Figure 102 the residuals of the in-sample forecast can be seen. The residuals are centered around zero, and there are a few slightly significant autocorrelation contributions. The p-value of the Ljung-Box test equals 0.002799 which means that the null-hypothesis of uncorrelated residuals is rejected and there is still some autocorrelation in the residuals left. In the histogram it can be seen that the residuals are centered around zero and the shape looks like a Bell-curve, but the figure is too spiked to have normally distributed residuals.

A.3 ARMA

In addition to the ARMA implementation in Section 4.3, the ARMA method is applied for storage 2 in 2019 and storage 4 in 2019. First storage 2 in 2019 is analysed to find a good fitting SARIMA model.

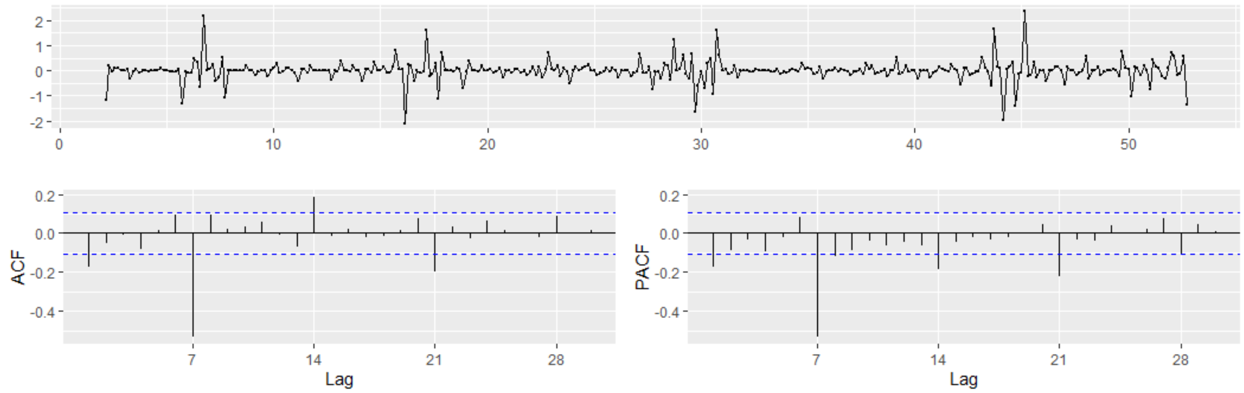

Figure 103: Time sequence plot, ACF and PACF of 2019 storage 2 after one time finite and one time seasonal differencing

In Figure 103, the time sequence plot, autoregressive function and partial autoregressive function are shown for the time series after one time finite differencing and one time seasonal differencing. In the autocorrelation function, lag 1 is above the significance level, and the first subsequent lags are not significant. This indicates that a suitable value for parameter q is 1, as seen in Section 3.5.2.
In the partial autocorrelation function, only lag 1 is above the significant level (except for multiples of 7). This indicates that 1 is a suitable value for parameter p, as seen in Section 3.5.1.
To estimate the values of parameters P and Q, the autocorrelation and partial autocorrelation functions of lag 7 and multiples of 7 can be used. For parameter Q, the autocorrelation function can be analysed. It can be seen that the contribution for lags 7, 14 and 21 are significant, and the contribution for lag 28 is not significant. Therefore 3 is a suitable value for parameter Q. In the partial autocorrelation function it can also be seen that lags 7, 14 and 21 are significant, and lag 28 is not. Therefore 3 is a suitable value for parameter P. It might be the case that some of the significant lags of multiples of 7 can also be caused by decaying behaviour, and that the best values of P and Q are lower than 3, but this will be investigated later. All together, the first model to try is SARIMA$(1,1,1)(3,1,3)_7$. This model is implemented in R and the results are given below.

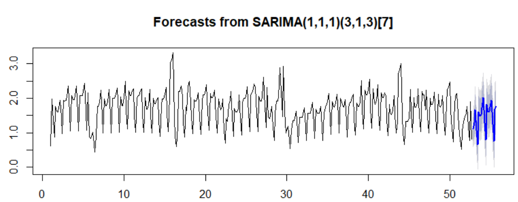

Figure 104: SARIMA$(1,1,1)(3,1,3)_7$ forecast 2019 storage 2

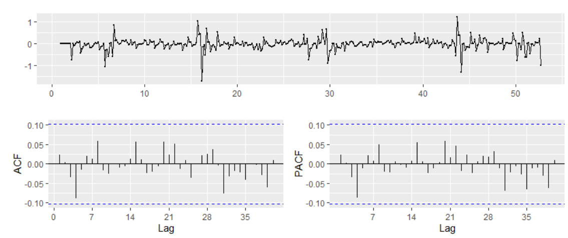

Figure 105: SARIMA$(1,1,1)(3,1,3)_7$ residuals 2019 storage 2

In Figure 104 the forecast is visualized (the blue line), while in Figure 105 the in-sample residuals are plotted, together with the autocorrelation function and the partial autocorrelation function. It can be seen that there are no significant autocorrelations left, which is an indication that the model fits the data well. On the other hand, the fact that not one autocorrelation lag is significant, might indicate that the model overfits the data, and less parameters can be used to model the data. Another indication of the quality of the model is Akaike's Information Criterion (AIC), which equals 110.36 for this model. To check if this is the best model, some variations can be applied and the residuals can be studied and the AIC values can be compared to find the best fitting model.
The next model that is fitted, is the SARIMA$(1,1,1)(2,1,2)_7$ model, shown in the figures below. The AIC

value is equal to 108.41, so according to the AIC value, this model is better to model the data. In Figure 107 it can be seen that there are still no significant autocorrelations, which is a desirable property, but it can be checked if this is not caused by overfitting.

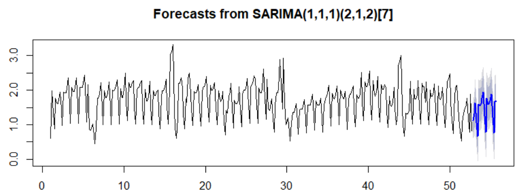

Figure 106: SARIMA$(1,1,1)(2,1,2)_7$ forecast 2019 storage 2

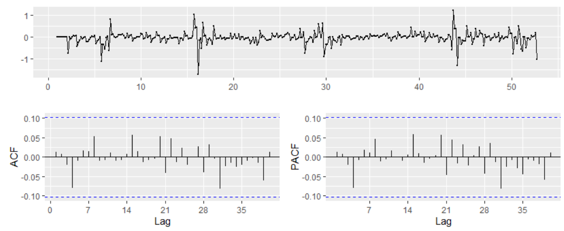

Figure 107: SARIMA$(1,1,1)(2,1,2)_7$ residuals 2019 storage 2

The SARIMA$(1,1,1)(1,1,1)_7$ model is implemented, and in Figure 109 the residuals are shown. This model has an AIC value that equals 112.67, which means that the SARIMA$(1,1,1)(2,1,2)_7$ model fitted the data better.

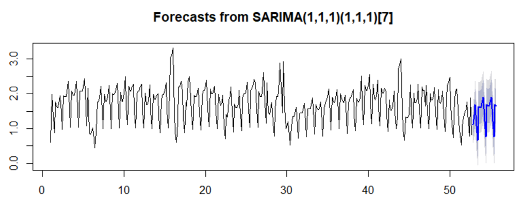

Figure 108: SARIMA$(1,1,1)(1,1,1)_7$ forecast 2019 storage 2

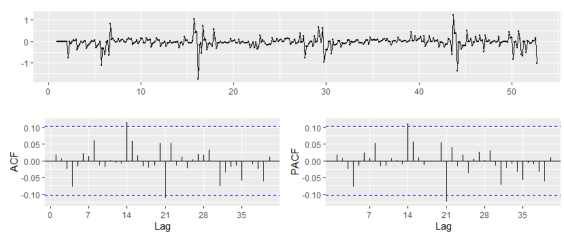

Figure 109: SARIMA$(1,1,1)(1,1,1)_7$ residuals 2019 storage 2

It can still be analysed if the SARIMA$(1,1,1)(1,1,2)_7$ or the SARIMA$(1,1,1)(2,1,1)_7$ model is an improvement compared to the SARIMA$(1,1,1)(2,1,2)_7$ model. The residuals are shown below.

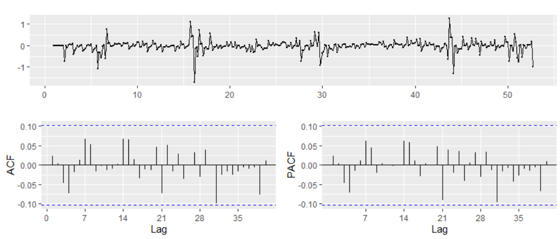

Figure 110: SARIMA$(1,1,1)(1,1,2)_7$ residuals 2019 storage 2

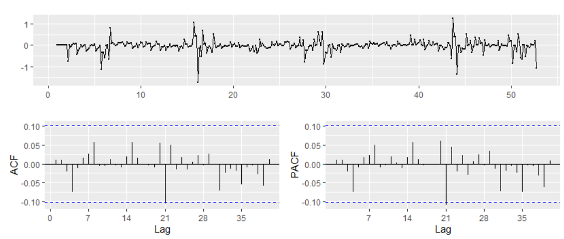

Figure 111: SARIMA$(1,1,1)(2,1,1)_7$ residuals 2019 storage 2

The AIC values are 108.65 and 110.72, so according to the AIC value, SARIMA$(1,1,1)(2,1,2)_7$ is the best model to forecast the sales for storage 2 in 2019.

The results are summarized in the table below.

Model	AIC value
SARIMA$(1,1,1)(3,1,3)_7$	110.36
SARIMA$(1,1,1)(2,1,2)_7$	108.41
SARIMA$(1,1,1)(1,1,1)_7$	112.67
SARIMA$(1,1,1)(1,1,2)_7$	108.65
SARIMA$(1,1,1)(2,1,1)_7$	110.72

Next storage 4 in 2019 is analysed to find a good fitting SARIMA model that includes trend and seasonality.

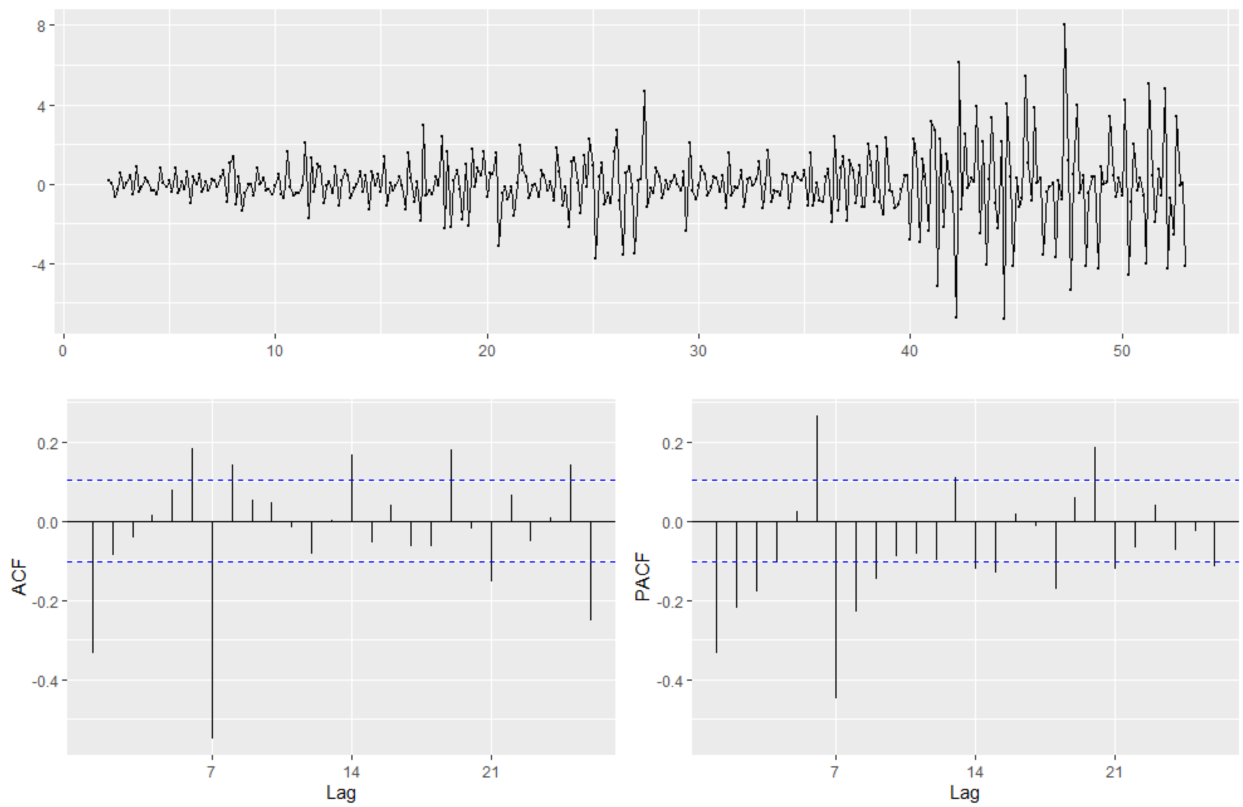

Figure 112: Time sequence plot, ACF and PACF of 2019 storage 4 after one time finite and one time seasonal differencing

In Figure 112, the time sequence plot, autoregressive function and partial autoregressive function are shown for the time series after one time finite differencing and one time seasonal differencing. In the autocorrelation function, lag 1 is above the significance level, and the first subsequent lags are not significant. This indicates that a suitable value for parameter q is 1, as seen in Section 3.5.2.
In the partial autocorrelation function, lag 1, 2 and 3 are above the significant level. This indicates that 3 can be a suitable value for parameter p, as seen in Section 3.5.1. After lag 1 a decaying behaviour can be seen, which indicates that the best value for p may be lower than 3, and this will be checked later.
To estimate the values of parameters P and Q, the autocorrelation and partial autocorrelation functions of lag 7 and multiples of 7 can be used. For parameter Q, the autocorrelation function can be analysed. It can be seen that the contribution for lags 7, 14 and 21 are significant. Therefore 3 is a suitable value for parameter Q. In the partial autocorrelation function it can also be seen that lags 7, 14 and 21 are significant, but 14 and 21 are only slightly above the significance level. Therefore 3 can be a suitable value for parameter P. It might be the case that some of the significant lags of multiples of 7 can also be caused by decaying behaviour, and that the best values of P and Q are lower than 3, but this will be investigated later.

All together, the first model to try is SARIMA$(3,1,1)(3,1,3)_7$. This model is implemented in R and the results are given below.

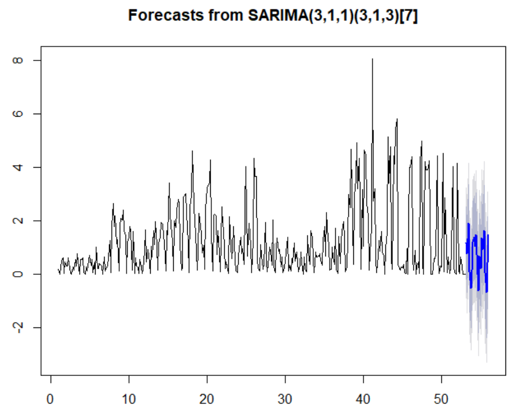

Figure 113: SARIMA$(3,1,1)(3,1,3)_7$ forecast 2019 storage 4

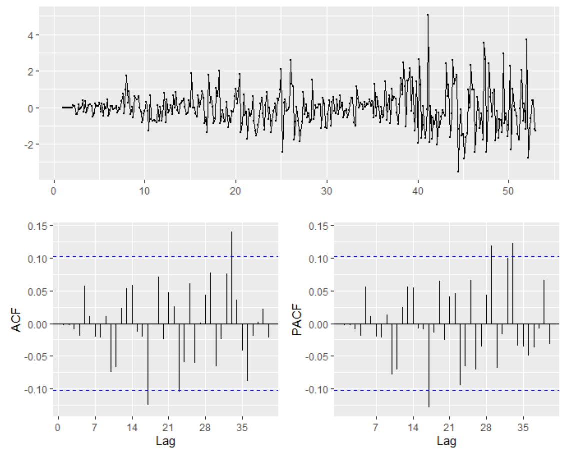

Figure 114: SARIMA$(3,1,1)(3,1,3)_7$ residuals 2019 storage 4

In Figure 113 the forecast is visualized (the blue line) with a 95% and 80% prediction interval, while in Figure 114 the in-sample residuals are plotted, together with the autocorrelation function and the partial autocorrelation function. It can be seen that there are no significant autocorrelations left, which is an indication that the model fits the data well. On the other hand, the fact that not one autocorrelation lag is significant, might indicate that the model overfits the data, and less parameters can be used to model the data. Another indication of the quality of the model is Akaike's Information Criterion (AIC), which equals 1086.28 for this model. To check if this is the best model, some variations can be applied and the residuals can be studied and the AIC values can be compared to find the best fitting model.
The next model that is fitted, is the SARIMA$(2,1,1)(2,1,2)_7$ model, shown in the figures below. The AIC value is equal to 1083.5, so according to the AIC value, this model is better to model the data. In Figure 116 it can be seen that there are still no significant autocorrelations, which is a desirable property, but it can be checked if this is not caused by overfitting.

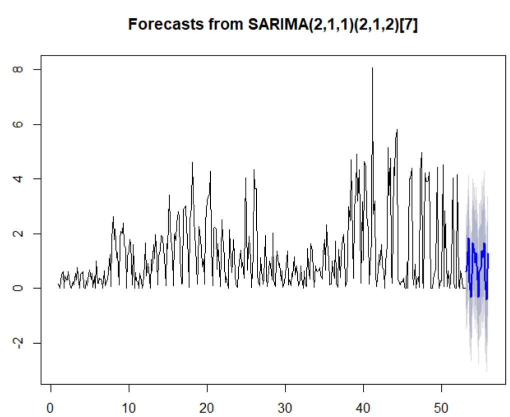

Figure 115: SARIMA$(2,1,1)(2,1,2)_7$ forecast 2019 storage 4

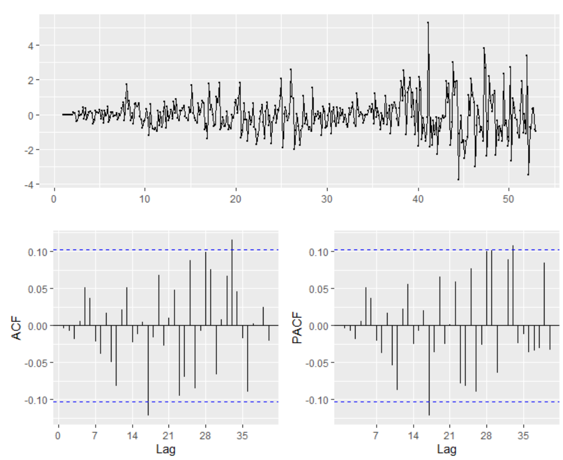

Figure 116: SARIMA$(2,1,1)(2,1,2)_7$ residuals 2019 storage 4

The SARIMA$(1,1,1)(2,1,2)_7$ model is implemented, and in Figure 118 the residuals are shown. This model has an AIC value that equals 1084.12, which means that the SARIMA$(2,1,1)(2,1,2)_7$ model fitted the data better.

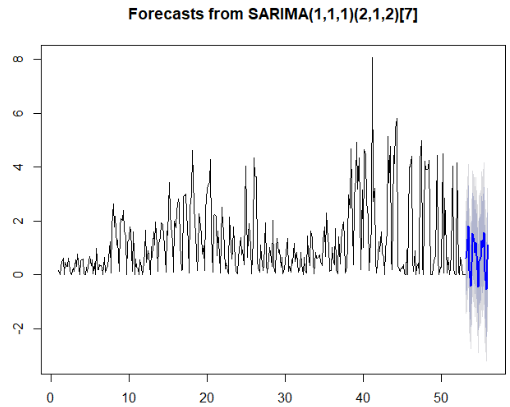

Figure 117: SARIMA$(1,1,1)(2,1,2)_7$ forecast 2019 storage 4

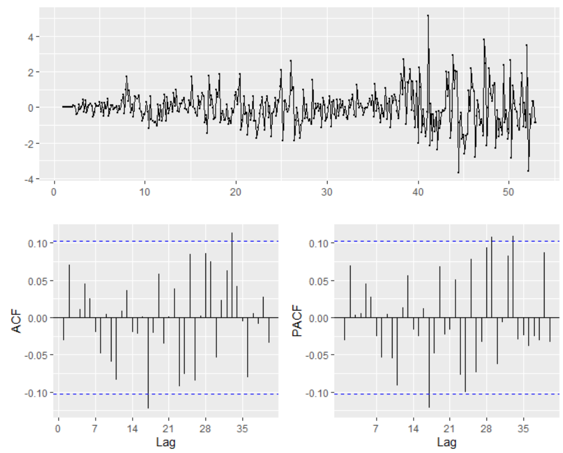

Figure 118: SARIMA$(1,1,1)(2,1,2)_7$ residuals 2019 storage 4

The SARIMA$(2,1,1)(1,1,2)_7$ model is implemented, and in Figure 120 the residuals are shown. This model has an AIC value that equals 1089.73, which means that the SARIMA$(2,1,1)(2,1,2)_7$ model fitted the data better.

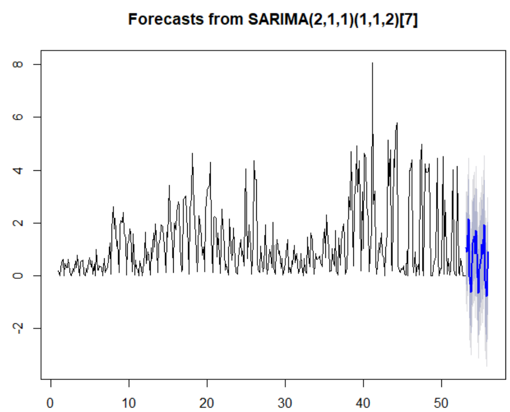

Figure 119: SARIMA$(2,1,1)(1,1,2)_7$ forecast 2019 storage 4

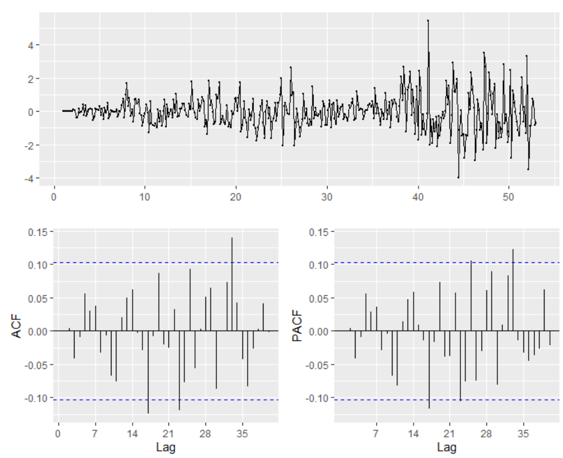

Figure 120: SARIMA$(2,1,1)(1,1,2)_7$ residuals 2019 storage 4

The SARIMA$(2,1,1)(2,1,1)_7$ model is implemented, and in Figure 122 the residuals are shown. This model has an AIC value that equals 1081.63, which means that this model fits the data best so far.

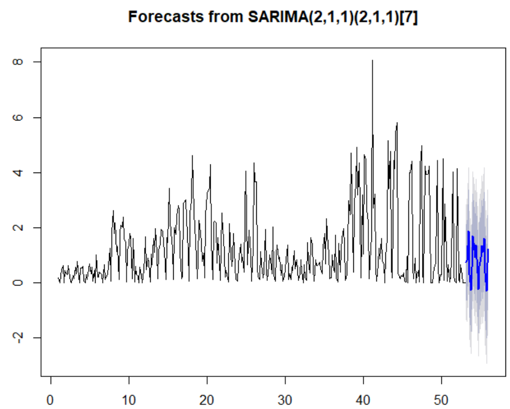

Figure 121: SARIMA$(2,1,1)(2,1,1)_7$ forecast 2019 storage 4

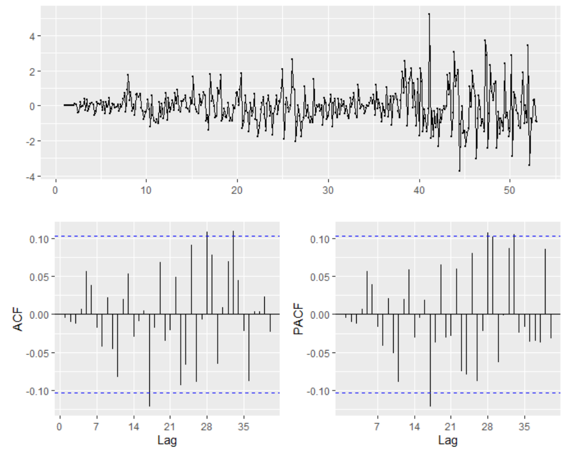

Figure 122: SARIMA$(2,1,1)(2,1,1)_7$ residuals 2019 storage 4

It can still be analysed if the SARIMA$(1,1,1)(2,1,1)_7$ or the SARIMA$(2,1,1)(1,1,1)_7$ model is an improvement compared to the SARIMA$(2,1,1)(2,1,1)_7$ model. The residuals are shown below.

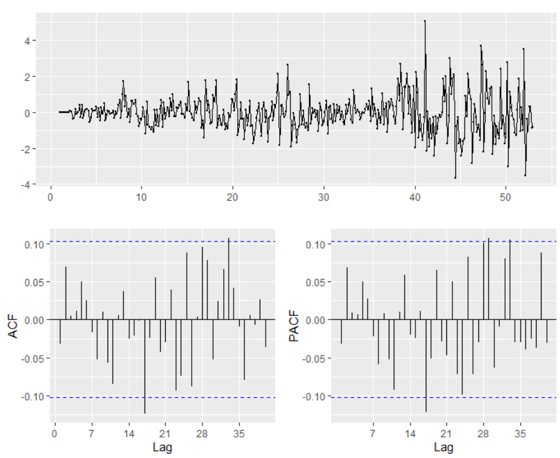

Figure 123: SARIMA$(1,1,1)(2,1,1)_7$ residuals 2019 storage 4

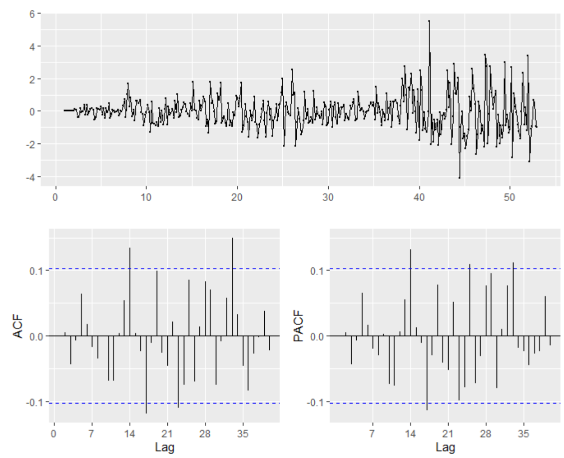

Figure 124: SARIMA$(2,1,1)(1,1,1)_7$ residuals 2019 storage 4

The AIC values are 1082.43 and 1091.34, so according to the AIC value, SARIMA$(2,1,1)(2,1,1)_7$ is the best model to forecast the sales for storage 4 in 2019.
The results are summarized in the table below.

Model	AIC value
SARIMA$(3,1,1)(3,1,3)_7$	1086.28
SARIMA$(2,1,1)(2,1,2)_7$	1083.5
SARIMA$(1,1,1)(2,1,2)_7$	1084.12
SARIMA$(2,1,1)(1,1,2)_7$	1089.73
SARIMA$(2,1,1)(2,1,1)_7$	1081.63
SARIMA$(1,1,1)(2,1,1)_7$	1082.43
SARIMA$(2,1,1)(1,1,1)_7$	1091.34

A.4 Box-Cox transformation

In addition to the implementation of the Box-Cox transformation in Section 4.3, the Box-Cox transformation is also applied for storage 1 in 2020 and storage 2 in 2019. First we will look at the results for storage 1 in 2020. After performing the Box-Cox transformation with $\lambda_2 = 2.7430812928761$ and the additive Holt-Winters exponential smoothing method in R, the following smoothing parameters are found.

$$\hat{\alpha} = 0.2986327$$
$$\hat{\beta} = 0.02021354$$
$$\hat{\gamma} = 0.03297379$$

All smoothing parameters are close to zero, which means that more weight is given to observations further in the past, and the model is called slow learning. The estimates for the level, trend and seasonal components are the following for the last time step T.

$$L_T = 2.943259720$$
$$T_T = -0.001911971$$
$$S_T = \begin{bmatrix} S_{T-s} \\ S_{T-s+1} \\ \vdots \\ S_T \end{bmatrix} = \begin{bmatrix} -0.436259967 \\ 0.133097173 \\ 0.079089028 \\ 0.030500293 \\ 0.160097835 \\ 0.193779566 \\ -0.198205161 \end{bmatrix}$$

In the figures below, the in-sample forecast and the residuals of the in-sample forecast of the Box-Cox transformed time series can be seen. The in-sample forecast is the forecast until time T (in red), plotted in one figure together with the observed values (in black). Note that the same data is used for both fitting and validation. When comparing different models in Section 4.6, also the out of sample accuracy will be analysed.

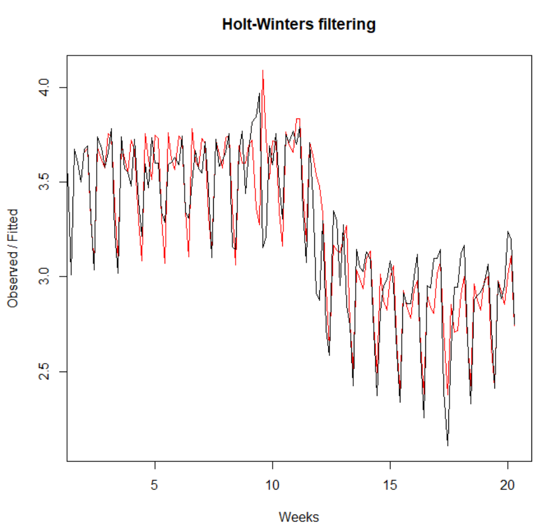

Figure 125: In-sample forecast for 2020 storage 1

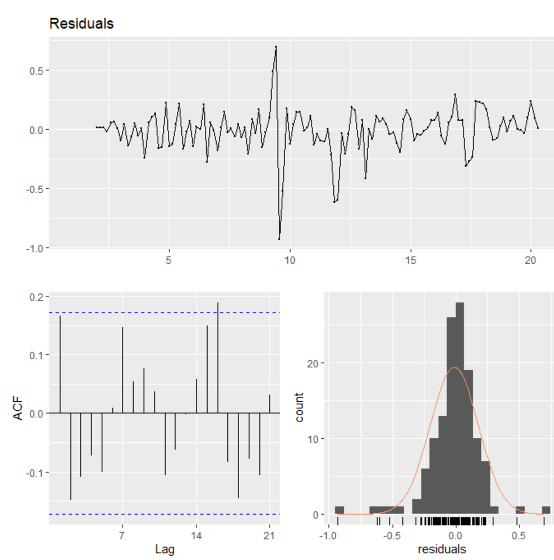

Figure 126: Residuals of in-sample forecast 2020 storage 1

In Figure 125 it can be seen that the forecast successfully captures the seasonal pattern, and it also captures the trend. The additive seasonality can be recognised in the forecast, because the seasonality amplitude stays

similar over time. The time sequence plot, autocorrelation function and a histogram of the residuals are shown in Figure 126. The residuals are centered around zero. The autocorrelation for lag 16 is a little above the blue significance line, so this could indicate that the residuals are not totally random and that some structure is left. The Ljung-Box test can be performed to check if there is overall significant autocorrelation left, with significance level $\alpha = 0.05$. The p-value of the Ljung-Box test equals 0.1424, which is not than α. This means that the null-hypothesis, that there is no significant autocorrelation left, is rejected. This might be a problem, but it can be checked if this is the case for multiple storages, or only for some storages.
Another aspect that has to be checked, is if the residuals are normally distributed. This can be checked using the Shapiro-Wilk test, with significance level $\alpha = 0.05$. The null-hypothesis of the Shapiro-Wilk test is that the data is normally distributed. when applying the test, an p-value of $2.25 \cdot 10^{-8}$ is found. This means that p is smaller than α, and the null hypothesis is rejected, so it is not assumed that the residuals are normally distributed. Fortunately, it is not necessary that the residuals are normally distributed, so the model can still be used. In the histogram it can be seen that the residuals are not normally distributed, but there is a centering around zero.

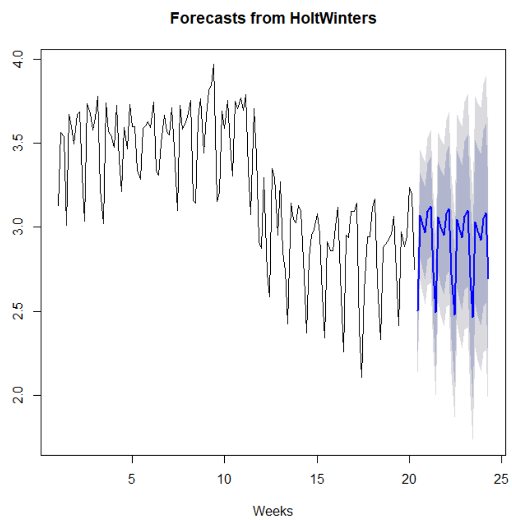

Figure 127: Holt-Winters forecast 2020 storage 1 on Box-Cox transformed scale

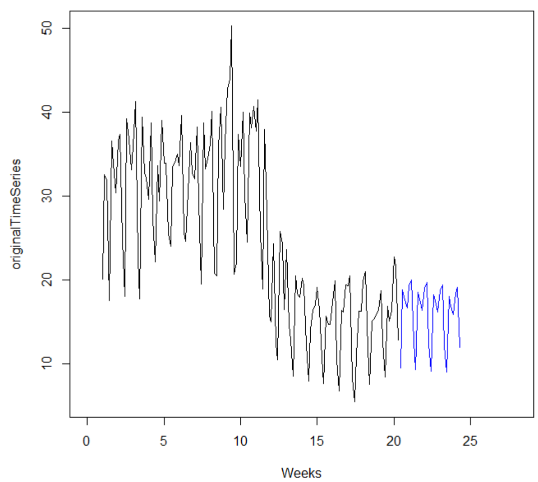

Figure 128: Holt-Winters forecast 2020 storage 1 on original scale

In Figure 127 the Holt-Winters forecast in the Box-Cox transformed scale can be seen. The blue line in the figure represents the forecasted values. To dark grey area is the 80% prediction interval for the forecast, and the light grey area is the 95% prediction interval for the forecast. In the forecast, the seasonality pattern can be seen, and there seems to be a small trend in the forecast.
In Figure 128 the forecast in the original scale can be seen, so this is the forecast after the inverse Box-Cox transformation is applied. It can be seen that the forecasts look quite similar, and since there are no zero-values or values close to zero in this time series, it can be evaluated later on if it is useful to use a Box-Cox transformation.

Next we will look at the results for storage 2 in 2019. After performing the Box-Cox transformation with $\lambda_2 = 0.224875250404688$ and the additive Holt-Winters exponential smoothing method in R, the following smoothing parameters are found.

$$\hat{\alpha} = 1$$
$$\hat{\beta} = 0.003787453$$
$$\hat{\gamma} = 5.328149 \cdot 10^{-16}$$

All smoothing parameters β and γ are close to zero, which means that for the trend and seasonality, more weight is given to observations further in the past, and the model is called slow learning. The smoothing parameter α is equal to 1, which means that the level estimate is fast learning, and only the last observation is used to determine the level component. The estimates for the level, trend and seasonal components are the following for the last time step T.

$$L_T = 0.619291134$$

$$T_T = 0.004053599$$

$$S_T = \begin{bmatrix} S_{T-s} \\ S_{T-s+1} \\ \vdots \\ S_T \end{bmatrix} = \begin{bmatrix} 0.064898954 \\ 0.051179348 \\ 0.096311126 \\ 0.188176870 \\ -0.086081664 \\ -0.445772009 \\ 0.131287374 \end{bmatrix}$$

In the figures below, the in-sample forecast and the residuals of the in-sample forecast of the Box-Cox transformed time series can be seen. The in-sample forecast is shown in red and the observed values are shown in black. Note that the same data is used for both fitting and validation. When comparing different models in Section 4.6, also the out of sample accuracy will be analysed.

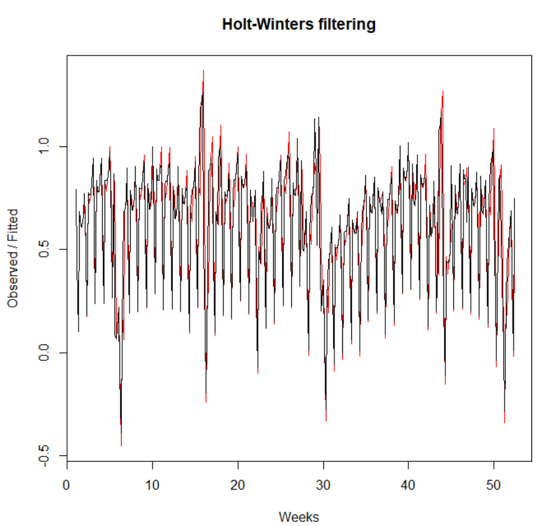

Figure 129: In-sample forecast for 2019 storage 2 on transformed scale

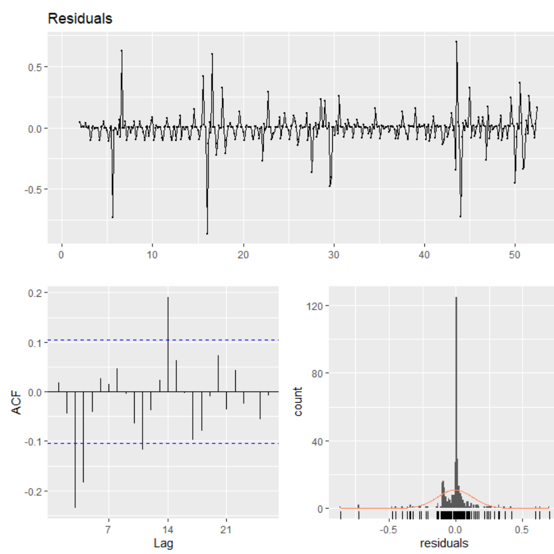

Figure 130: Residuals of in-sample forecast 2019 storage 2

In Figure 129 it can be seen that the forecast successfully captures the seasonal pattern, and it also captures the trend. The time sequence plot, autocorrelation function and a histogram of the residuals are shown in Figure 130. The residuals are centered around zero, and there is a very big spike at zero. The autocorrelation for lags 3, 4, 11 and 14 are above the blue significance line, so this could indicate that the residuals are not totally random and that some structure is left. The Ljung-Box test can be performed to check if there is overall significant autocorrelation left, with significance level $\alpha = 0.05$. The p-value of the Ljung-Box test equals $7.852 \cdot 10^{-5}$, which is not than α. This means that the null-hypothesis, that there is no significant autocorrelation left, is rejected. This might be a problem, but it can be checked if this is the case for multiple storages, or only for some storages.

Another aspect that has to be checked, is if the residuals are normally distributed. This can be checked using the Shapiro-Wilk test, with significance level $\alpha = 0.05$. The null-hypothesis of the Shapiro-Wilk test is that the data is normally distributed. when applying the test, an p-value of $2.2 \cdot 10^{-16}$ is found. This means that p is smaller than α, and the null hypothesis is rejected, so it is not assumed that the residuals are normally distributed. Fortunately, it is not necessary that the residuals are normally distributed, so the model can still be used. In the histogram it can be seen that the residuals are not normally distributed, but there is a centering around zero.

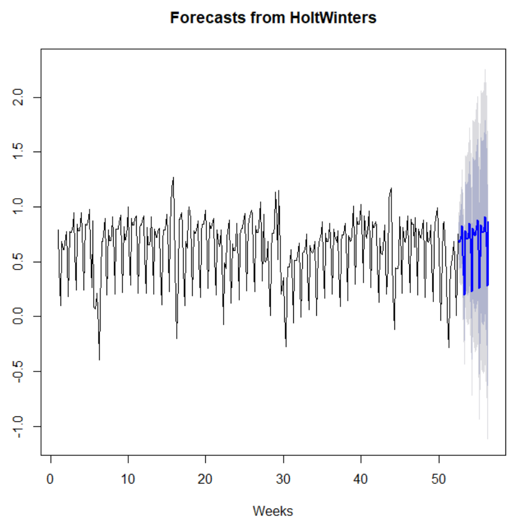

Figure 131: Holt-Winters forecast 2019 storage 2 on transformed scale

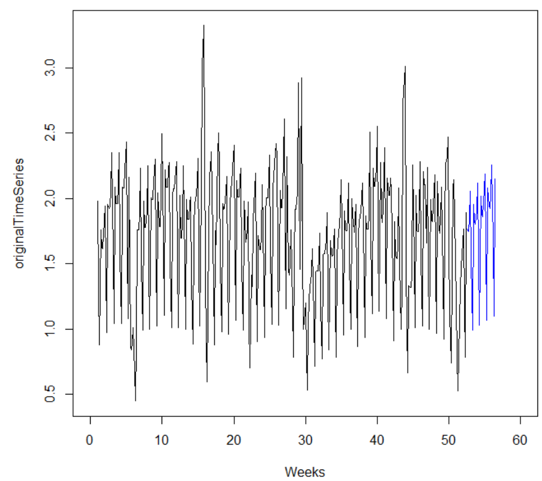

Figure 132: Holt-Winters forecast 2019 storage 2 on original scale

In Figure 131 the Holt-Winters forecast in the Box-Cox transformed scale can be seen. The blue line in the figure represents the forecasted values. To dark grey area is the 80% prediction interval for the forecast, and the light grey area is the 95% prediction interval for the forecast. In the forecast, the seasonality pattern can be seen, and there seems to be a small trend in the forecast.
In Figure 132 the forecast in the original scale can be seen, so this is the forecast after the inverse Box-Cox transformation is applied. It can be seen that the forecasts look quite similar but there is a bigger trend contribution in the forecast on the original scale. Since there are no zero-values or values close to zero in this time series, it can be evaluated later on if it is useful to use a Box-Cox transformation.

A.5 R code: Simulation and new methods

The code of the simulation and the comparison of the different new methods can be found using the following URL: https://github.com/DionneHeuvelman1/GraduationProject.