INVENTORY MANAGEMENT AT VELD KOELTECHNIEK B.V.

Bachelor thesis

Wossink, D. (Dirk Wossink, Student B-IEM)
University of Twente/Veld Koeltechniek

Company supervisor
S. Hoog-Antink (Sander)
Logistics manager

First supervisor University of Twente
R.M. van Steenbergen MSc (Robert)
Faculty of Behavioural Management and Social Sciences (BMS), Industrial Engineering and Business Information Systems (IEBIS)

Second supervisor University of Twente
Dr. D.R.J. Prak (Dennis)
Faculty of Behavioural Management and Social Sciences (BMS), Industrial Engineering and Business Information Systems (IEBIS)

Preface

Dear reader,

You are about to read my thesis regarding the inventory management at Veld Koeltechniek B.V., where the aim of this research is to provide the company with advice regarding their inventory and leave them something that they can work with.

This research could not have been finished without the help of some people. First of all, I would like to thank Robert van Steenbergen for being my supervisor. He helped me with important points for setting up the research and gave me some tips regarding the writing of the thesis and the planning. The feedback I obtained from every meeting was always very useful and helped me consider things from a different perspective as well. The feedback was critical, but I could always get in contact with him whenever I had a question or I felt like I needed more feedback. Besides that, I felt his support and interest in this research. Again, thank you very much for all your help during my research. Secondly, I would like to thank Dennis Prak for being my second supervisor and providing crucial feedback during the last phase of this research. His feedback was very important to get to the end result of this thesis. Without the feedback of you both, this thesis could not reach this final version as it is.

Finally, I would like to thank Sander Hoog-Antink, logistics manager at Veld and my supervisor at the company. Although he was busy often, he was always supportive, and very interested in my research. If I needed something or I needed to speak to someone, I could always contact Sander to help me. Besides that, he helped me get on the right track to the final result.

I hope you enjoy reading my thesis.

Yours sincerely

Dirk Wossink

July 2023

Management summary

Veld Koeltechniek B.V. is a company specialising in all sorts of refrigeration technologies. Part of their work is the aftercare of the installations. The service department operates several vans that mechanics use to go to jobs where service is needed. The management problem is the high costs of the inventory of Veld. The inventory of the service vans also contributes to these costs. Based on this we have formulated the main research question of this thesis:

How can we improve the inventory levels of the items within the service vans of the company, where we want to decrease the cost but still maintain the service level?

Veld uses standard lists that state what the inventory of their vans must be. These lists show for every type of van what items should be in there and what the inventory level should be. To improve the inventory levels of the service vans, we created new standard lists for their catering, retail, and industry service vans. These are the types with the most vans and also the ones with the most sales. The first step was to create a decision flowchart, which is used to determine which item should end up on the standard list for each van type. Certain items should be excluded from the standard list. These filters were determined by Veld since they know what type of items should be on the standard lists. By using this decision flowchart to determine which items should end up on the standard list we could already reduce the number of items on the standard list from the catering vans by 75%, from the retail vans by 61%, and for the industry vans we obtained a reduction of 68%.

The next step was to set certain reorder points and order-up-to-levels for the remaining products on the lists. We found that Veld uses a (0, S)-policy for the resupplying of their vans. In their way of resupplying we found a CSL of 80%, which lead to high stockout costs. The order costs were also high since the order-up-to-levels are low, meaning they will need to resupply often. Besides that, the order-up-to-levels S were determined by gut feeling. We created an (s, S)-policy, for which we computed the reorder point and the order-up-to-level for each item, which were based on formulas found in the literature. This led to a decrease of 98% in stockout costs and 93% in order costs. The holding costs increased by 163%, which is a large increase in percentage, but the increase in costs is small compared to the decrease in the other costs. The total costs of the current three standard lists together are assumed to be € 182,367.00, but with our (s, S)-policy and the CSL based on an ABC classification we did, we managed to decrease the total costs by 92%, which came down to a total cost of € 14,617.63.

The (s, S)-policy was evaluated using different general CSLs. The company proposed to use a general CSL of 98%. The total cost of using this service level was € 14,862.80, slightly higher than the (s, S)-policy with the CSL based on the ABC classification. Using a CSL of 99% proved to decrease the total costs even more than the CSL of the ABC classification. The total costs with this CSL were € 14,597.22. But when using a CSL of 99%, one will get more expected inventory.

These steps were implemented in VBA, where we created a dashboard that the company can use. The dashboard contains a couple of buttons that represent the abovementioned steps. The first button represents the decision flowchart, where the VBA code selects the right items for every type to end up on the standard lists. Pressing the next button enables the VBA code to do the computations for the reorder points and the order-up-to-levels of the items on the lists. The last button is used to evaluate each generated list based on certain costs.

This research was done with some assumptions and limitations. We had to make assumptions regarding the costs, the demand, and within the order policy of Veld. Unavailable data limited our research. We could not take into account the size and weight of the items. Besides that, we did not know the location to which an order had to be sent. Both of these could influence our order cost since the larger packages come with extra costs and packages to Belgium come with extra costs. The use of the CSL is a simplification of this research. We did not have data available that tells us if a job is done with an item or not, which thus meant we could not use the job fill rate as an indicator that might be more suitable.

With these assumptions, we managed to answer our main research question. Using a (0, S)-policy, where we maintain the 80% CSL, we managed to decrease the total costs by around 89%. We even showed that we could decrease the total costs and increase the service level in an (s, S)-policy, with a general CSL of 99%, where we decreased the total cost by almost 92%.

Table of contents

Chapter 1: Problem definition

This thesis will describe the research conducted at the logistical department of Veld Koeltechniek B.V. in Groenlo. This research investigates the possible reduction of inventory costs for Veld. The first chapter of this research covers the problem definition of this research. The company Veld Koeltechniek B.V will be introduced plus the problems they are facing. A visualization of these problems is shown. One of these problems is chosen as the core problem of this research. The steps for solving this problem will be described along with a description of the contents per chapter.

1.1 Introduction of the company

Veld Koeltechniek B.V. is a company based in Groenlo that was founded in 1960. The company is specialized in all sorts of refrigeration technology. They import, modify, develop, install, and sell all sorts of solutions for cooling and refrigeration. Refrigeration installation companies, wholesalers, kitchen specialists, supermarkets, and companies within the food industry are part of their customers. These customers are linked to one of the business units they operate in: wholesale, retail, food industry, industry, and service.

Part of their work is the aftercare of the installations. The service department operates several vans that mechanics use to go to jobs where service is needed. Different types of vans exist but these are not the same as the business units they operate in. The types of vans Veld uses are: catering, retail, industry, retail/catering, and IBS. The latter are vans for which Veld provides inventory but they do not belong to Veld. IBS is a Dutch abbreviation for In-Bedrijf-Stelling.

1.2 Management problem

A management problem is a problem referring to an issue or challenge arising within an organization. The problem owner, Veld Koeltechniek B.V., has a management problem. This problem has to do with the company's inventory management, which they want to improve. The problem that the company is currently facing is increasing inventory costs. The company saw that, especially during the Corona period, these costs have risen. The inventory of the company comprises the inventory within its warehouse as well as the inventory within each individual van. The company can track the total inventory of the company, which has a value

Confidential

The company would like to reduce this number, as well as the cost of the inventory. We define the management problem of the company as follows:

Veld Koeltechniek B.V. wants to lower its inventory costs

1.3 Problem identification and problem cluster

In interviews with people within the company, it was made clear that there are multiple problems underlying this management problem. The management problem is defined as the rise of the costs of the inventory. The costs of the inventory have risen because the inventory was and is still growing. The piling of the inventory is caused by several reasons. Firstly, all equipment and materials needed for a job come from the warehouse in Groenlo, while some materials could also be brought to the job site directly. This ordering policy for these items can be seen as not an ideal one.

Secondly, there is a lack of knowledge on the use of various reports. The company tracks its inventory by various reports but it does not know which reports are important and how they can use them for an improvement analysis. There are (i) products with a minimum stock level that are not sold for years. Due to the fact that it has a minimum stock level, these products will stay part of the company. There are also (ii) products that have minimal to no demand, but since they are not sold, they are not seen within the system often which thus means that the inventory levels of these products will not be changed. The inventory levels of these products are not updated and are too high, which thus leads to unnecessary amounts of inventory.

The above-mentioned problems already aim at a sub-optimal inventory, which leads us to the last problem. That is the inventory policy of the service vans. The company uses a standardized list of what the inventory should be in each van for new mechanics and for vans that come back to Groenlo for a periodical counting of inventory. These inventory lists are not updated every once in a while and will thus lead to piling of inventory that is not needed for new mechanics. This causes a sub-optimal inventory for the service vans.

These problems all redirect to the same problem which is the problem with the supervision of the inventory of the company. This problem is large since inventory has multiple aspects to it and providing supervision to all these aspects is not possible for this research. Therefore we chose the action problem regarding the inventory of the service vans as the core problem of this research. The identified problems are visualized in a problem cluster which is shown in Figure 1.1.

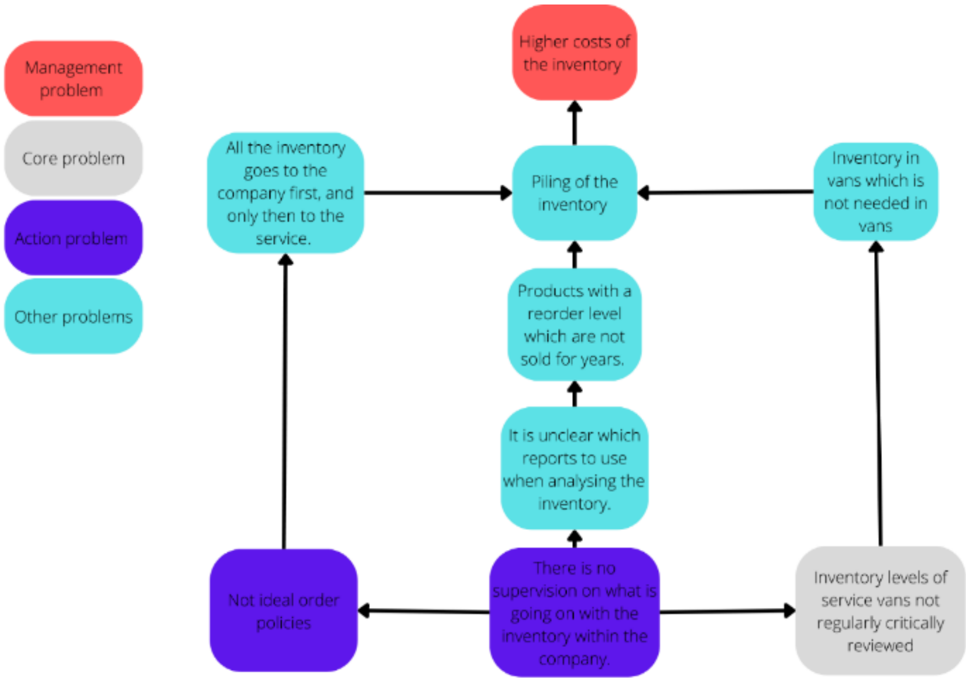

Figure 1.1: Problem cluster

1.4 Main research question

Defining research questions helps to create the scope of this research. The answer to the main research question solves the core problem. The sub-questions are related to the main research question. They break down the main research question and will answer a specific part of the main research question. This section will discuss the main research question, the sub-questions will be discussed in Section 1.6.

Main research question

From the core problem, the main research question of this research can be identified. Based on this problem, the following main research question is formulated:

How can we improve the inventory levels of the items within the service vans of the company, where we want to decrease the cost but still maintain the service level?

The problem of the standard lists for the service vans will be tackled together with the not optimal inventory levels. Working on this question will lead us to update the inventory levels of the service vans. Giving items an order level with some reasoning behind it will lead to substantiated and better levels and a probable reduction in costs rather than determining these levels by gut feeling.

1.5 Solution to the problem

The company wants a solution that creates clarity and certain set rules for the standardized lists. As part of improving the inventory for the service vans, the company wants to update these standardized lists for the most common van types. These lists indicate for each type what items should be in the van and with what level. Inventories could be improved by considering individual products and individual vans, but the company wants to stick to the existing use of lists to keep more overview and supervision. To create the desired clarity, a decision flowchart will be created which makes it clear for everyone in the company why a certain item is on the list and that they can check if they can put an item on that list by following that flowchart. For certain van types, we will first create new and updated standardized lists.

The decision flowchart will decide which items will end up on the lists but it also needs to be decided with what quantity they will end up on each list. We will give items a general service level based on a certain classification. This classification will be done using an ABC classification. An ABC classification is an inventory classification technique that splits a large number of inventory items into categories. Through this classification, inventory items are managed in an efficient way (Kaabi, 2022).

With this knowledge in mind, the main deliverable for the company is a structured inventory policy that is implemented in VBA, which can be used by a dashboard. Veld can run the VBA code by pressing buttons on the dashboard. The VBA code will then generate the standard lists, showing the reorder points and the order-up-to-levels of the items on the standard lists, and will calculate the total costs of the generated lists.

1.6 Structure

The structure of this research is based on the MPSM (Heerkens & Van Winden, 2017). We first defined the problem, after which we moved on to the problem-solving approach. From there we analyse the problem, come up with solutions, and eventually implement and evaluate them. Not all the chapters correspond to one step from this method but every step is followed. This gives the following structure to this research:

- Chapter 2: The literature study

This chapter aims to give us knowledge that will be central to the rest of this research. It consists of two parts. We introduce some definitions based on literature in the first part. The second part is focused on previous research in which similar problems are discussed. A similar problem is the repair kit problem. This part provides us with input on what we can use in our final model and which aspects we should consider. Besides that, we look at the previous use of the ABC classification in research regarding inventory management. The question that will be answered is:

What aspects of the literature regarding an ABC classification and the repair kit problem can we use?

- Chapter 3: The analysis of the current situation

In this chapter, we will analyse the current process of the supply of service vans. The policy that Veld uses for this process will be identified. We will analysis the current standard list and the actual inventory levels. This analysis is used to further expose the state of the problem, to set assumptions for our model, and to later compare with the outcomes of the proposed methodology. The question central in this chapter is:

What is the current inventory policy of the company and what is the state of the current inventory, in terms of value on the standard list and the actual inventory?

- Chapter 4: The development of the inventory policy.

Chapter 4 contains the methodology for our solution for the company. The elements that are needed for the solutions will be described in this chapter, which includes the decision flowchart, ABC classification, and the method behind the dashboard. Certain choices and assumptions will be explained. In the decision flowchart, we will explain why certain elements were chosen. The use of the ABC classification will be explained by using the literature. Finally, this chapter will cover the tool we created for Veld to generate new standard lists. This will cover the following question:

How can we develop a new inventory methodology for the service vans?

- Chapter 5: The outcomes

A chapter with the results of our new methodology. We can compare our proposed solution to the current situation and then show the change in inventory costs if the company decides to use the proposed solution. We will also analyse our proposed solution with various CSLs to see the impact of the service levels. This answers the following question:

What are the changes in costs and the impact on the service levels of using the proposed methodology instead of the current lists and policy?

- Chapter 6: The conclusion

This chapter will give the overall advice for the company. The main findings from this research will be given in this chapter and ultimately the answer to the main research question. Our model gives a proposed solution to the standard lists but there are some limitations and simplifications to it that leave space for further research, which we will also discuss.

Chapter 2: The literature study

In this chapter, we discuss relevant scientific literature on topics relevant to this research. We aim to incorporate relevant knowledge discussed in this chapter in our methodology. The study is twofold. Firstly, we introduce concepts regarding inventory management which are necessary for the remainder of this research in Section 2.1. Following a literature study, where we find relevant scientific literature regarding certain topics. The steps of this study will be explained in Section 2.2. Sections 2.2.1 and 2.2.2 will then provide the outcome of the literature review, after which we will describe our findings regarding the ABC classification, and the repair kit problem in Section 2.3. This chapter will answer the following research question:

What aspects of the literature regarding an ABC classification and the repair kit problem can we use?

2.1 Review policy

This subsection sets the definitions that will be used later throughout this thesis and from which it is essential to know what they mean and what is meant with it in the context of this research. This information is based on the book by Chopra & Meindl (2015). We will identify the components for a continuous review policy and the formulas for the reorder points and order-up-to-levels.

Periodic and continuous review policies

In inventory management, we can identify two types of inventory policies to review inventory: periodic and continuous review policies. A periodic review policy reviews the inventory levels at fixed intervals, also known as the review period *R*. The continuous review policy monitors the inventory level continuously (i.e. the review period is equal to zero). Central in these policies are the following variables:

- Review period – *R*
- Lead time – *L*
- Reorder point – *s*
- Order quantity – *Q*
- Demand – *D*
- Order-up-to-Level – *S*

There are various examples of each type of policy. The (s, S) -policy, where an order is placed if the inventory level drops below the reorder point *s* with a variable quantity *Q* so that the inventory level will be the order-up-to-level *S*, a fixed level the inventory should be after ordering. In an (s, Q)-policy one also orders if the inventory level drops below the reorder point *s*, but the order always has a fixed quantity *Q*. The main difference between them is thus the fixed *S*, where one orders a quantity *Q* to get to that level, and the fixed *Q*, where one always orders a fixed quantity. The difference between a continuous and a periodic review policy is that a periodic review policy will check at certain time intervals if the inventory level is below the reorder level *s*.

Economic Order Quantity (EOQ)

We can use the formula for the Economic Order Quantity (EOQ), to determine an optimal quantity of goods in order to minimize inventory holdings and ordering costs. It takes into account factors such as the demand rate, ordering cost, and holding cost per unit. The formula calculates the ideal order quantity that balances the cost of carrying inventory with the cost of placing orders. This formula is as follows:

$$Q = \sqrt{\frac{2 * D * S}{h * C}}$$

where:

Q = Optimal order size

D = Demand per time unit, where the EOQ assumes deterministic demand

S = Ordering costs

h = Holding cost per time unit as a fraction of the unit costs

C = Unit costs

Safety inventory and service level

Safety inventory is a type of inventory that is held in excess of expected demand to protect against stockouts due to unexpected fluctuations in demand or supply chain disruptions. It serves as a buffer that can help maintain customer satisfaction and avoid lost sales. In case of uncertainty in demand, the safety inventory acts as a buffer to cover unexpected increases in demand. The calculation for the safety stock is used in the computation for the reorder point. This is the following formula:

- $ss = z*\sigma_L$

In this formula, σ_L denotes the standard deviation of the demand during the lead time. The z-score in the safety stock formula is linked to the cycle service level (CSL) of a company. The CSL is the probability of not having a stockout in the review period. The CSL is normally expressed in a percentage, which can be calculated over time by dividing the number of cycles not having a stockout divided by the total number of cycles. If demand during a stockout is backlogged, one can use the following formula to determine the optimal cycle service level:

- $CSL = 1 - (\frac{HQ}{DC_u})$

where:

H = hC, the cost of holding one unit for one unit of time

Q = Optimal order size

D = Demand per time unit

C_u = Understocking cost

Using the formula for the safety stock, we can also compute the order points *s* and the order-up-to-level *S* in an (s, S)-policy:

- $s = D_L + ss$
- $S = Q + s$

2.2 Literature Review

The following paragraphs follow from literature about previous research. In order to develop a suitable approach for this research, it will be helpful to find information that can be used. The focus will be on the use of an ABC classification and the repair kit problem, which is a problem similar to this research's problem. For each topic we explain the definitions, discuss relevant works and developments and then decide what could be used for the methodology.

2.2.1 ABC-classification

The ABC classification is a method to assign a value to items that fall into a specific group. It classifies items in groups based on how important they are for the company, where items in category A are the most important items. The traditional ABC classification is carried out using the Pareto principle or the 80/20%- rule. (Khanorkar & Kane, 2022)

Following the Pareto principle, it is often the case that only 20% of the products account for 80% of the revenue. Then it could be that items in category B generate 15% of revenue and category C items only 5 %. These percentages are not that strict but are mentioned by Mantilla et al (2022). Although the items in category A are items which are the most critical, these are most of the times chosen with a low service level. High service levels will lead to high safety stock and thus high costs (Winston, 2004). Therefore, it can be favourable to set a high service level of items in category C since increasing the safety stock for this group is relatively cheap.

Flores & Whybark (1986) were the first to consider a joint criteria classification. They proposed to use a joint criteria matrix when considering two criteria in a classification. Examples of criteria that one can use are revenue and annual usage. This matrix then consists of nine different cells. These cells are AA, AB, AC, BA, BB, BC, CA, CB, and CC. This means that a product can end up in one of 9 different cells and can thus be classified in more detail. This type of classification matrix is useful for this research.

Douissa & Jabeur (2020) stated that there were four main classes when using an ABC classification. There are classification approaches based on mathematical programming, AI and meta-heuristics, Multi-Criteria Decision Making (MCDM), and hybrid techniques.

The first linear programming (LP) model to solve the multi-criteria ABC classification was introduced by Ramanathan (2006). This model used a weighted additive function as its objective function, which is used to combine the performance of various inventory items with the corresponding weights of the criteria to produce a normalized score. The downside to this model is that items can perform badly in some significant criteria but still end up in a good class due to the full compensability of the model. The cause of this is the significantly high value on other criteria.

To overcome this downside, Zhou and Fan (2007) proposed a classification based on two LP models. One model maximized the best score for an inventory item and the other minimized the worst scores. Afterwards they proceeded with a convex combination of the two scores to generate an overall score for each item.

Another extension to the model of Ramanathan is the Ng-model from the research by Ng (2007). The extension made sure that the decision-maker could add constraints expressing an ordinal ranking on the criteria weights. With some simplifications, the Ng-model has shown that it can be solved without any linear optimizer since it only uses one equality constraint.

A recent article by Atakay et a.l. (2022) used an ABC classification for spare parts in a service sector company. They constructed an ABC-VED matrix, an example of a joint criteria classification. First, they classified the ABC side of the matrix, based on the demand and the costs, where 8 % of the items ended up in class A, 20% in class B and 72% in class C. Next they did a VED analysis based on an Analytical Hierarchy Process (AHP). A VED analysis is an analysis where a classification is done based on the criticality of items. In this case, V are the *Vital* items and thus the most important items, E are *Essential* items, and D are *Desirable* items. The AHP had as output a certain weight for a group. These weights represented the percentage of items belonging to that category. The boundaries for each class have been determined after which items could be placed into one. This matrix is shown in Figure 2.1.

	V	E	D
A	AV	AE	AD
B	BV	BE	BD
C	CV	CE	CD

Figure 2.1: An ABC-VED matrix

2.2.2 Repair kit problem

The problem regarding the inventory of the service vans is not a new phenomenon in the world of inventory management. The problem of determining which spare parts need to be in a service van and in what quantity is called the repair kit problem. Throughout the years a lot of research has been done regarding the repair kit problem, but not all is relevant to this research. Before going to a job, a mechanic of Veld does not know which items are necessary to complete that job. Veld currently does not have the IT infrastructure to consider advanced demand information. The demand in our research is therefore imperfect and thus is literature regarding advanced demand information not relevant (e.g.. Rippe and Kiesmüller, 2023).

The repair kit problem was initially explored by Smith et al. (1980), Graves (1982), and Hausman (1982). Their research focused on tours consisting of a single job, with independent part demands and a maximum of one unit required for the repair. Later work extended this problem to encompass dependent demands and multiple units of a part within a single job, as well as spare machines and budget constraints (Mamer and Smith 1982, March and Scudder 1984, Mamer and Smith 1985, Mamer and Shogan 1987). Brumelle and Granot (1993) presented a unified approach to address different formulations of the single job repair kit problem.

The multiple-job repair kit problem, where a tour involves several customers visiting before the repair kit is restocked, was first examined by Heeremans and Gelders (1995). They formulated a more generic version of the repair kit problem and proposed a heuristic to find solutions for this multiple-job problem, using the probability that none of the jobs on a tour is broken as a service criterion. Teunter (2006) presented a more general model formulation for the multiple job problem, where all required parts available in the repair kit are left with the customers, regardless of whether the job can be completed. He developed a cost model and a service model with a job fill rate constraint, and a greedy heuristic to solve the problem.

Bijvank et al. (2010) discussed the same problem but showed how the job fill rate can be computed when spare parts are only taken from the repair kit when all parts necessary for the repair are available in the correct quantity. Saccani et al. (2016) and Prak et al. (2017) introduced replenishment costs to the cost model in their recent contributions on the multiple job repair kit problem. Saccani et al. (2016) assumed fixed replenishment costs per delivery to the service technician and determined optimal frequencies for these deliveries. Prak et al. (2017) provided a multi-job, multi-item, and multi-unit with positive lead time, where they assume an independent demand for the items. This research provided a model which computes the job fill rate for certain reorder levels and order-up-to-levels.

2.3 Conclusion from the literature

This literature was conducted in twofold where there are a couple of points that help us answer the question:

What aspects of the literature regarding the same topic as this research can we use?

The first part of this chapter was about setting the terms. This led to the following important points:

- We found formulas to calculate the safety stock, reorder points, and the order-up-to-levels for a continuous policy, a policy where the inventory is continuously monitored.
- The formula for the EOQ has been formulated, which we will use the determine the order quantity.
- The formula for the CSL has been formulated, which we will use in our ABC classification to determine general service levels for each category.

The following points are concluded from the search to the ABC classification which we will use for our research in Chapter 4:

- The Pareto principle, where items in category A take 80% of the revenue, while only consisting of 20% of the total items, category B covers 15% of the revenue, and lastly category C covers the 5% left.
- A joint criteria matrix can be used to classify based on more than one criterion.
- A normalized score can be produced based on the performance of various inventory items, according to the weight of certain criteria.
- An ABC-VED matrix takes the revenue and the essentiality of a product into account when constructing the matrix. We saw that the essentiality in the article by Atakay et al. (2022) was determined by an AHP. In our research we will define the essentiality of a product based on its sales in a year.

Lastly, we analysed the repair kit problem literature. From this we conclude:

- The repair kit problem looks at the combination of items necessary for certain jobs. A solution to solving the repair kit problem is to minimize the costs of either overstocking an item or understocking an item.
- The reviewed literature took the job fill rate as a measurement for their solutions. As we described in Section 2.1, we take into account the CSL. There is no data available that

tells us if a job is done with an item or not, which is necessary for the job fill rate. Using CSL is also a simplification for this research since the computations are less complex.

- This research will eventually provide a list of items with reorder points and order-up-to-levels. This list of items can be seen as the combination of items that are used for certain jobs. Minimizing the costs of either overstocking an item or understocking an item will be done by using the CSL, which represents the trade-off between these two costs.

Chapter 3: The analysis of the current situation

In this chapter, we analyse the current state of the inventory of the company. In Section 3.1 we will focus on the elements of an inventory review policy which we have defined in Section 2.1. In Section 3.1 we will see what Veld does regarding their service vans. Section 3.2 will show an analysis of the current standard lists, which can be used as a benchmark to which we can compare the final solution. In Section 3.3 we will show the actual inventory at a certain moment in time. We finalize this chapter with conclusions in Section 3.4. This chapter answers the question:

What is the current inventory policy of the company and what is the state of the current inventory, in terms of value on the standard list and the actual inventory?

3.1 Inventory in vans

This section will explain what type of vans Veld uses, what the standard lists are for the company, and how the vans are resupplied with a certain delivery system.

Van types

Part of the company´s inventory is in its service vans. These vans are used for the installation or the reparation of their products elsewhere. These vans contain items that are used for these processes, also known as spare parts. There are more than 60 vans and multiple types of vans. The types of vans correspond with the sector in which they are deployed and are specialized in service for that type. These sectors are not the same as the business units the company operates in. The van types that the company has are catering, retail, industry, retail/catering, IBS, and NKT. In-bedrijf-stelling (IBS) are vans that do the starting of certain installations. This means they do not need an inventory of spare parts. Niet-Koel-Technische (NKT) vans are vans that do service in not-refrigeration technologies. These are special types of vans, for which the company does not want inventory and are thus out of the scope of the solution methodology.

Standardized lists

The inventory of a service van is determined by a standard list. This standard list contains certain items with a value to it for the type of van it should be in. This value is an order-up-to-level S. Not all the van types have standard lists. There is a list for catering, retail, industry, retail/catering, and IBS vans. When an item is not on that list, it is an item that should not be in the van. A new mechanic that starts working at Veld gets a van with an inventory. The inventory of this van is based on the standard list, according to the type that this mechanic is going to work in. These lists were made a couple of years ago and are thus outdated. Besides that, the order-up-to-levels of the items might also not be optimal since this value is based on gut feeling.

Resupplying policy

All of the usages are collected in the company's Enterprise Resource Planning (ERP) system, which is Microsoft® AX. The people at the warehouse track the inventory of every van. If an item in a van has a certain S, but the physical stock is zero, it needs to be resupplied. The company will not only resupply a van for one item but it will fill the van with more items. For this they look at the needs of an item. The need is defined as the difference between the physical stock and the order-up-to-level of an item. If an item has a physical stock of two and an order-up-to-level of five then the need of this item for this van is three. If the need is larger than the physical stock of that item, then this item will also be resupplied to the order-up-to-level. For a single item we identify an (s, S)-policy, where the s is always zero. This way of working can be done, but in that case the company ignores the demand during lead time and safety stocks. However, Veld does not order every item individually, instead other items in a van will be ordered if its physical stock is equal to (S/2)-1

Delivery

Bringing an item from the warehouse in Groenlo to the mechanic happens via a delivery system. Whenever an item is below the minimum stock level, a van will be resupplied. This happens via a night delivery. A van is supplied from the items that the company has within the warehouse. The vans however do not come back to the warehouse every day or every week. The vans stay in the area that they are working in at that moment. A night delivery will collect the items that are needed for the vans at the warehouse in Groenlo and will send this to a central depot located in Breda. From there the items will be delivered to the van. If a van is out of stock on day 1 then at the end of day 1 this special delivery system will collect the item at the warehouse in Groenlo and take it to the depot in Breda. From Breda to the van also takes 1 day of processing, so on day 3 the van will be resupplied. That means that the van needs to bridge 1 day without an item. This delivery system is quite reliable, so we assume the standard deviation in lead time is zero and thus we define a lead time of 1.

The costs for these shipments are known and are fixed. The cost that it takes for an item to go from Groenlo to Breda is € 38.12. Then the cost of delivering the items from Breda to the van has a fixed cost of € 13.57. It is a fixed cost that is irrespective of the distance the delivery has to travel. It should be noted that this is the cost of delivering a package of items within the Netherlands. For delivering a package from Breda to Belgium there are higher costs. Then a delivery has a fixed cost of € 17.01. There are also extra costs if a bigger package is needed. Such a package will cost € 4.21 extra.

All this information can serve as input for our methodology. First of all, we identify a continuous policy, where an item can be resupplied to an order-up-to-level if its level is zero, or if an item has a level lower than S/2 and another item in that van has a level of zero. This is according to an (s, S)-policy, where we denote the *S* as the level to which it needs to be resupplied and the small s as the level at which it needs to be resupplied. The delivery system makes sure that the vans will be resupplied quickly so that the lead time is fixed and is always 1. We assume that this happens since in practice some latency might occur. The cost of this delivery is fixed. The cost of taking an item from Groenlo to a van is € 38.12 + € 13.57 = € 51.69. It is mentioned that there are extra costs for delivering in Belgium and larger packages, but we neglect these since these are rare to happen and because we cannot take them into account with the data we have available. We do not consider the location of where a van needs to be resupplied and thus it is assumed that this is always in the Netherlands. The size of an item is also not taken into account. We do not have this data available so we cannot say if a larger package is necessary. Therefore we assume it is never necessary.

3.2 Analysis of the standardized lists

For the analysis of the standardized lists, we looked at the most recent lists. It should be noted that we analyse the standard lists of the IBS and catering/retail vans, while the company wants to scrap these.

The standard list is a list of items that should be in a service van. Every item has its order-up-to-level that is also stated on that list. Five types of vans have such a list, for which we thus analysed the current state. For the analysis we took the value of each item and multiplied that by the *S* to get the value of the standard list when for example a new mechanic starts at Veld.

There are currently 65 vans that are linked to such a type. The values of these standard lists are shown in Table 1. The table shows what the inventory value of that list is if a van is full with all the items and the order-up-to-levels on the corresponding standard list, how many of every van type there is, and the total inventory value if all the vans of a certain type were filled with every item on that list.

Van type	Inventory value	Quantity	Total
Industry	€ 8,313.74	19	€ 157,961.03
Catering	€ 10,246.78	18	€ 184,442.12
Retail	€ 10,597.79	22	€ 233,151.45
Catering/Retail	€ 12,169.70	1	€ 12,169.70
IBS	€ 10,918.13	5	€ 54,590.66
Total			€ 642,314.95

Table 1: Value of the current standard lists

The standard lists contain a total value of € 642,314.95, but that is not equal to the expected average inventory value. If every item was reordered at level zero then the expected average inventory level would be equal to S/2. However only one out of five items ordered will reach that level at a resupply, so in 20% of the cases $\frac{S+0}{2}$ holds. The other 80% is ordered when the inventory level is below half of the order-up-to-level (i.e., S/2-1). For these items $\frac{S+\left(\left(\frac{S}{2}\right)-1\right)}{2}$ holds. Based on these assumptions, the expected average inventory value for each type can be seen in Table 2.

Van type	Expected level of inventory
Catering	€ 4,038.51
Retail	€ 4,238.70
Industry	€ 3,227.73
Catering/Retail	€ 4,643.71
IBS	€ 4,089.01

Table 2: Expected inventory value

3.3 Actual inventory of the vans

Confidential

3.4 Conclusion from the current situation analysis

This chapter formed the benchmark to which we compare our results. Starting with the ordering process. An (0, S)-policy was identified where the company would always re-supply a van if the inventory level of an item with an order-up-to-level reached the inventory level 0. The company would then also order other items. The items they would also order would reach the inventory below half of their order-up-to-level. We assume that they would order five items in total, so that means that only one item reaches inventory level zero and the others do not. The resupply will then happen via a special delivery which will go via a main depot in Breda to the vans, where a fixed cost is attached to this supply. The identified policy can be used, but then the company neglects the safety stocks and the demand during lead time. In our methodology we will therefore propose an (s, S)-policy, where we thus take into account a safety stock. This policy will most probably reduce the stockout costs the most.

An analysis of the standard lists and the actual inventory has been made. In Chapter 5 we compare the outcomes of our proposed solution to this analysis. The proposed standard lists have an expected inventory level, which we can compare to the current standard lists and the actual inventory. The actual inventory at a certain moment in time and the sales from 2022 told us that a large proportion of the inventory at that moment was not sold from any van at all. Even if we looked at each type there was a proportion of inventory not sold from any van of that type, indicating space for improvement when deciding which items are necessary for the service vans.

Chapter 4: The development of the inventory policy

This chapter will show the methodology behind the solution of the new standard lists for the service vans of Veld. The decision flowchart will be explained in Section 4.1. The choice of certain branches will be substantiated. Section 4.2 will show how the ABC classification was done. In Section 4.3 we will discuss the method for the inventory policy, where we will discuss some assumptions and formulas used in the VBA code. Lastly, Section 4.4 will discuss how we analysed every list using certain formulas. This chapter aims to answer the question:

How can we develop a new inventory methodology for the service vans?

4.1. Decision flowchart

A wish from the company was that the selection of an item being on the standard list is more substantiated. Everyone within the company should know why an item is on the standard list or why it is missing. For this, we will construct a flowchart in consultation with the company. This flowchart will filter the relevant items for the standard lists. There are certain exclusion criteria for when an item does not belong on a standard list and certain inclusion criteria for why an item does belong on the list. These criteria have been decided in interviews with employees of the service department, who know what characteristics are important. These criteria are the following:

Must-have articles

The must-have articles are defined as the articles that are in the vans because it was agreed on with the customers to be in the van. That might happen if a customer has an installation for which an item is necessary for the service of that installation. At this moment, there are not any of these items, but it has been in the past and it could also occur in the future. Therefore the company wants this to be an exclusion filter in the flowchart. The company will track these items differently, which means they are not necessary for the standard list.

Floor stock

Floor stock is another group of items for which the company said that they do not belong on the standard list. Floor stock is a group of items that generally have low value and for which the mechanics can decide if they need certain items and in which quantities they want them. Nuts and bolts are examples of floor stock items. There is a list on which the mechanics can order these items. These items should therefore not end up on the standard list. Floor stock items have that characteristic attached to them in the system. For this model we use these items as the floor stock items. The company has to check in the system if all these items are rightly classified as floor stock or if it needs to be updated.

Refrigerant

Refrigerant is an item that the company said should not be on a standard list. It is a liquid item and is kept track in terms of litres. As with floor stock items, refrigerant is tracked differently and, therefore the company stated that this should not be an item that belongs on the standard list.

Value

The company stated that they did not want high-value items to be on the standard list. The company stated that this limit should be €200.00. Items with a value higher than that value will be filtered out and will not end up on the standard list. The literature on the repair kit problem tells us that we should minimize the cost between an item in the van and not selling it, and not having an item in the van and thus not finishing a job. Therefore some advice for the company would be to change this limit to a value where the cost of missing an item is less costly than the holding costs.

This threshold is the value when the costs of running out of stock during an order cycle (i.e., understocking costs) are lower than the costs of holding an item in inventory for one order cycle (i.e., overstocking costs). The costs of running out of stock are unknown. These costs can be seen as a penalty. We set this penalty cost at € 58.63, which is the understocking cost. For the overstocking cost we assume a holding cost of 20% if an item is in a van for a whole year. We thus divided the percentage of the holding cost by the number of cycles an item has in a year to get the holding cost of an order cycle.

Using this we find:

Overstocking costs = (20% * product value) / number of order cycles

Understocking costs = € 58.63

Below is a small table mentioning possible number of order cycles per year and the maximum product value for which the overstocking costs are cheaper than the understocking costs:

Order cycles:	Product value
5	€ 1,465.75
10	€ 2,931.50
15	€ 4,397.25
20	€ 5,863.00
25	€ 7,328.75

Table 5: Possible order cycles versus maximum product value

From this table, € 1,465.75 is the lowest product value, but there is no item Veld sell from their vans that is worth so much. Therefore we can conclude that the limit of € 200.00 Veld stated is good to use for the solutions since there are only 25 items that were sold in 2022 that have a value higher than this.

Sales

The last node we look at is how much a certain product is sold in the last year. It has been decided that if an article is not sold in a year from a certain type of van, it does not need to be on the standard list of that type. The order-up-to-level for that item will then be zero. Another group contains articles that are rarely sold. This is a group of articles where an item is sold only once or twice a year from a certain type of van. These articles will be put on another generated list where the head mechanic later can decide if they want it still on the standard list or not. The last group will be the group that will be classified and is thus the group for which we need to determine s and S. These items will be discussed in Chapter 4.2. A visualization of this flowchart can be seen in Figure 4.1

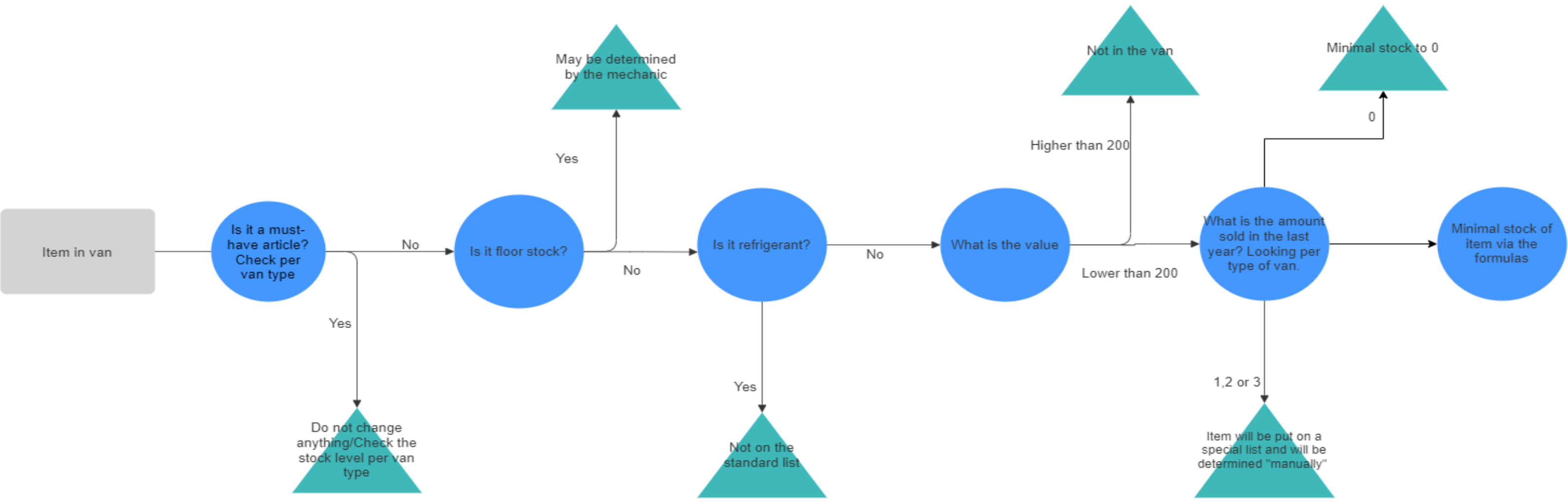

Figure 4.1: Decision flowchart

4.2 ABC-classification

After going through the decision flowchart, items will need to be on the standard list of a type. Determining with what quantity and thus the inventory level of a certain item will be done by using the formulas we defined in the literature and an ABC classification of the items. We will start with determining the boundaries for the categories. If an item has a value of X and is Y times sold in a year it will end up in a certain category, for example BE. For categories A, B, and C we take the average value of the holding costs for each category. We multiply this by the factor of the V, E, or D category to get a value for the overstocking cost for that joint category. Based on these costs we determine a CSL for each category with the correct Z-score to it.

We will use an ABC-VED matrix as Atakay et a.l.(2022) also did, but we do not determine the essentiality based on an AHP but on the sales of an item. In their matrix, the ABC noted the value of an item and VED the essentiality of an item. Based on assigning 80%, 15%, and 5% of the products to, respectively, A, B, and C, we determine the required percentiles of the product value and the sales volume as thresholds for the categories. The 20th percentile of the product value is € 37.60, which means all the products with a value above € 37.60 will be classified in category A. The 95th percentile of the product value is € 12.07. Products with a value below this will end up in category C. So products with a value between € 12.07 and € 37.60 end up in category B. A total of 756 products are considered in this analysis, where 204 would end up in category A, 198 in category B, and 354 in category C. That is around 26% for both categories A and B and almost 50% for category C.

At the VED side of the matrix we look at the sales of the products. With this we mean the amount of times an item has been sold. A total of 8973 sales were done in 2022. For the VED categories we did the same as the ABC categories, where category A is now resembled by V, category B by E, and category C by D. The 20th percentile of the sales is determined at 11 sales per year. The products with sales higher than 11 cause 80% of the total sales, while it is only 143 products (19%) that have this. These products end up in category V. Category D products have sales lower than 2. This is 5% of the total sales and 32% of the product. Then items with sales between 2 and 11 end up in category E, which is around 48% of the products (367). This leads to the following matrix, where between brackets the amount of items is shown that end up in this category:

Value↓ | Sold→	Category V 80% = 11+	Category E 15% = 2-11	Category D 5% = 2-
Category A 80% = €37.60+	(28)	(97)	(79)
Category B 15% = €12.07-€37.60	(44)	(91)	(63)
Category C 5%= €12.07-	(71)	(179)	(104)

We will use the formula for the CSL as we have defined in Section 2.1 to determine the service levels for each category. We already know the demand *D* and the order quantity *Q*. The understocking cost for every item is the same, which is € 58.63. The last thing we need to know is the overstocking cost. We take the average overstocking cost for each category to finally determine an average CSL for every category.

We start by taking the average value of each category on the ABC side of the matrix. This can be translated into an average overstocking cost per category. The average overstocking costs per year is 20% of the average value of each category. This leads us to the following overstocking cost for every category:

Category	Average value	Overstocking cost per year
A	€ 119.79	€ 23.95
B	€ 22.76	€ 4.55
C	€ 4.24	€ 0.80

Table 6: Average value per category

We will then use the VED side of the matrix to get to a specific overstocking cost for each of the nine categories. To get to a specific overstocking cost for each category, we multiply the average overstocking cost per year of a category by a factor corresponding to category V, E, or D. This factor is based on the percentage of products that should end up in this category. Category V should have 5% of the product, which thus gives it a factor of 0.05. The same reasoning holds for the categories, where items ending in category E get a factor of 0.15 and items in category D get a factor of 0.8.

We can now calculate the average overstocking cost for each category. Using this, the demand for each item, the order quantity of each item, and the understocking costs, we can determine a CSL for each item, after which we take the average of that category. The results of this are shown in Table 7.

Category	Average value	Factor	Average overstocking costs	Average CSL	Z-score
AV	€ 23.95	0.05	€ 1.19	96,87%	1.86
AE	€ 23.95	0.15	€ 3.59	96,26%	1.78
AD	€ 23.95	0.80	€ 19.16	93,94%	1.54
BV	€ 4.55	0.05	€ 0.22	99,13%	2.37
BE	€ 4.55	0.15	€ 0.68	98,90%	2.29
BD	€ 4.55	0.80	€ 3.64	98,16%	2.08
CV	€ 0.80	0.05	€ 0.04	99,54%	2.60
CE	€ 0.80	0.15	€ 0.12	99,38%	2.50
CD	€ 0.80	0.80	€ 0.64	99,19%	2.40

Table 7: Average CSL per category

If we fill in the Z-scores we get our ABC-VED matrix as shown below. The values of this table can be seen as reasonable. Looking at the ABC side of the matrix, one would expect that cheaper items will need higher safety stocks since missing out on these items is more costly in relation to the penalty cost than a more expensive item is. For the VED side, one would expect that items that are sold frequently have a higher safety stock than less frequently sold items, since the demand during the lead time is higher. In our matrix this is also the case so that makes it reasonable.

Value↓ | Sold→	Category V 80% = 11+	Category E 15% = 2-11	Category D 5% = 2-
Category A 80% = €37.60+	1.86	1.78	1.54
Category B 15% = €12.07-€37.60	2.37	2.29	2.08
Category C 5%= €12.07-	2.60	2.50	2.40

4.3 Inventory policy

We developed a method for a new inventory policy of Veld. The methodology is implemented in VBA and can run on the background of the dashboard to generate new solutions. This model has certain assumptions and limitations to it. We assume that:

- The demand in 2023 is equal to the demand in 2022. There is no forecasted demand for the upcoming period. Thus we base our calculations on the data of 2022.
- The stockout cost during an inventory cycle is € 58.63 irrespective of the number of units short.
- The data in the system is correct. That means that the mechanic writes off an article correctly, it ends up in the system correctly and it is retrieved from the system correctly. Not only the sales data is assumed to be correct but also the data from the floor stock. A list with items that have floor stock as characteristic in AX has been used as the source for the filtering of the floor stock items. The assumption is that this list is correct and up to date.

- The lead time is always 1 day. The explanation of this process has been done previously, but in reality some delivery delay may occur, but this is something we neglect in the model. The standard deviation of lead time is thus zero.
- Demand is stationary and follows a normal distribution.

With these assumptions, we thus coded two different inventory policies. We have coded a (0, S)-policy, as Veld currently uses this one, and an (s, S)-policy. To determine the reorder points and the order-up-to-levels we have used the following formulas:

- Q = EOQ = $\sqrt{\frac{2*D*S}{h*C}}$
- ss = z*σ_L
- s = $D_L + ss$
- S = $D_L + ss + Q$

In the (0, S)-policy, the order-up-to-level is equal to Q, since the reorder point *s* is zero. σ_L and D_L are determined based on historical data. The data we use are the sales per day, from which can determine the standard deviation during lead time and also the demand during lead time, where we divided the total demand of a product for a certain van type by 250, which in this case represents the number of working days in a year. σ_L and D_L might thus be different between the types of vans and we thus take into account that the right numbers are used for the right type. The Z-scores are based on the ABC classification from Section 4.2. The Z-scores can be varied based on what CSL one wants their standard list to be.

4.4 Analysis of the inventory policy

The next step is the calculation of the costs of the standard lists. We evaluate the lists, with a reorder level s, and an order-up-to-level S, based on its holding costs, stockout costs, and order costs.

Holding costs

The expected yearly holding costs of the lists will be based on the average inventory level of a certain item. For the average inventory level we use the following formulas:

Average inventory level =$0.2*\frac{S}{2}+0.8*\frac{S+((\frac{S}{2})-1)}{2}$ for the (0, S)-policy of Veld

Average inventory level = $\left(s+\left(\frac{S-s}{2}\right)\right)-D_L$ for the (s, S)-policy

Using the average inventory levels we can then calculate the expected yearly holding costs of each list. We do this by the following formula:

$$Holding\ Costs = average\ inventory * (0.2 * product\ value)$$

Stockout costs

The yearly stockout costs of the list are determined by the item´s CSL and the number of cycles it goes through yearly. These costs are fixed costs for when an item reaches the inventory level of zero. We have defined the stockout costs earlier to be € 58.63. In a (0, S)-policy a stockout happens when an item reaches a physical inventory level of zero and the demand during lead time is zero. In the way Veld works only one of the five items that is ordered will reach level zero, so the chance of a stockout happening in an (0, S)-policy is 20%.

The chance of a stockout happening in an (s, S)-policy is determined by the CSL, where every cycle there is a chance of going out of stock. This gives the following formula:

$$C = (1 - CSL) * 58.63 * \frac{Annual\ demand}{EOQ}$$

Order costs

The yearly order cost is the amount of times an item should be ordered yearly, based on its annual demand and the EOQ. The order costs are € 10.32, where we assume the company orders five different items at once, where the quantity for every item of that five is determined by the formula of the EOQ. Therefore we divided the total order cost by five.

Dashboard

The dashboard is the tool that Veld will use to help determine the standard lists based on an inventory policy. In the VBA code behind it, we have implemented all the elements discussed in this chapter. Figure 4.2 shows what the dashboard looks like. Pressing the buttons on there will run the VBA code.

Starting with the first button, which represents the decision flowchart in Section 4.1 It puts the items on the standard list for each type. The second button runs the VBA code that calculates the reorder points and order-up-to-levels. The formulas for this have been discussed in Section 4.3. The last button is used to calculate the costs, as we described earlier in this section.

It is made in such a way that the user of this dashboard only has to use the buttons and change parameters to whatever they would like to see. There is a section with certain parameters, where one can change the maximum value for an item to end up on the standard list, which is currently set at € 200.00, and the minimum amount of sales of an item, which is currently set at 2. Besides that, one can also change the Z-scores for each category. The code does not need to be touched, meaning that the user of this dashboard will only need to change the cell value of the parameter to get different results.

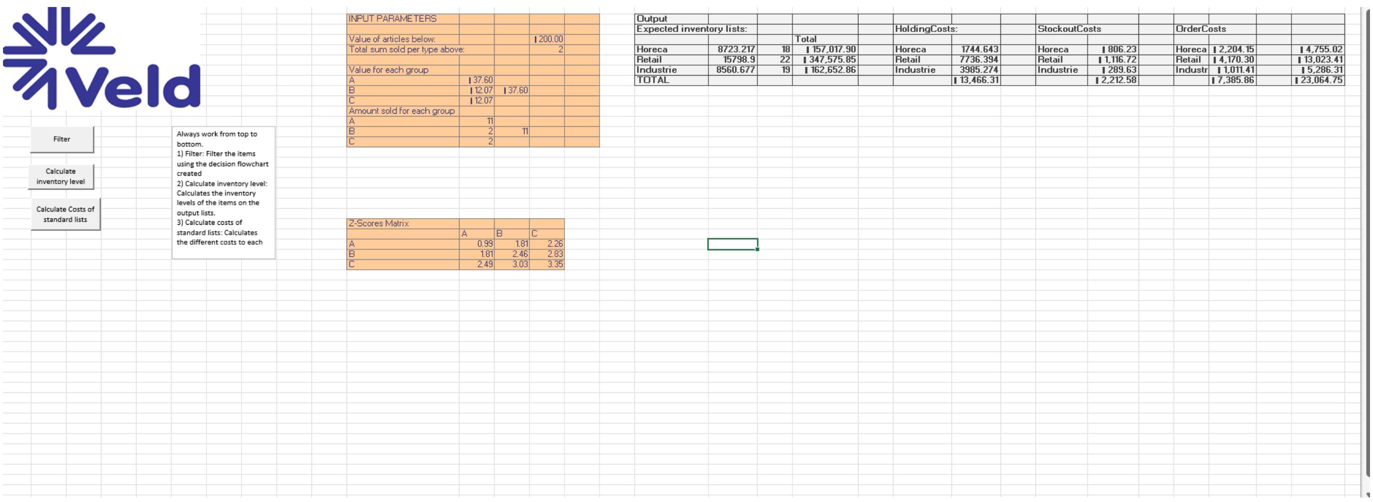

Figure 4.2: Interface of the dashboard

Chapter 5: The outcomes

This chapter will cover the outcomes of our methodology. We evaluate the generated standard lists of different scenarios. We will evaluate the lists based on the expected inventory levels and certain costs as we described in Section 4.3. This evaluation will show the difference between the current lists and our proposed new policies. This chapter will aim to answer the following question:

What are the changes in costs and the impact on the service levels of using the proposed methodology instead of the current lists and policy?

5.1 Expected inventory level

We will analyse the outcomes of the methodology based on five different scenarios. These are the following:

- Scenario 1: The current standard lists of Veld, where the *S* values were already determined and the items were not filtered by our flowchart.
- Scenario 2: The current standard lists of Veld, where the *S* values were already determined, but the items are filtered by our flowchart.
- Scenario 3: A new standard list where we use the (0, S)-policy Veld currently uses. The items are filtered by our flowchart and *S* is equal to *Q*. The CSL in this scenario is 80%.
- Scenario 4: A new standard list where we use an (s, S)-policy. The items are filtered by our flowchart and every item has a CSL of 98%, a service level proposed by Veld.
- Scenario 5: A new standard list where we use an (s, S)-policy. The items are filtered by our flowchart and the CSL of the items is based on the ABC classification in Section 4.2.

First, we evaluate the expected inventory value of the scenarios. For the expected inventory value of the current standard list of Veld we will use the analysis of Section 3.2. The expected inventory value per scenario is reported in the tables below, based on the calculations as described in Section 4.3.

Scenario 1	Expected inventory value	Quantity of items on this list
Catering	€ 4,038.51	380
Retail	€ 4,238.70	357
Industry	€ 3,227.73	280

Table 8.1: Expected inventory value of scenario 1

Scenario 2	Expected inventory value	Quantity of items on this list
Catering	€ 1,454.62	79
Retail	€ 1,860.82	97
Industry	€ 703.62	47

Table 8.2: Expected inventory value of scenario 2

Scenario 3	Expected inventory value	Quantity of items on this list
Catering	€ 9,585.39	94
Retail	€ 16,853.46	139
Industry	€ 8,176.34	90

Table 8.3: Expected inventory value of scenario 3

Scenario 4	Expected inventory value	Quantity of items on this list
Catering	€ 8,732.34	94
Retail	€ 15,913.44	139
Industry	€ 8,542.29	90

Table 8.4: Expected inventory value of scenario 4

Scenario 5	Expected inventory value	Quantity of items on this list
Catering	€ 8,757.87	94
Retail	€ 16,021.31	139
Industry	€ 8,610.82	90

Table 8.5: Expected inventory value of scenario 5

We can see that our proposed solutions end up with a higher expected inventory value than there currently is. The reason for this is the *S*-levels of the old list, which are much lower compared to the new proposed lists. This can be concluded from the comparison of scenario 3 to scenario 1. Scenario 3 shows the highest expected level of inventory, which has the same policy as scenario 1. The difference is that scenario 3 has fewer individual items compared to scenario 1, which thus should mean that the inventory levels of the items in scenario 3 must be higher and that the order-up-to-levels of these items have also increased.

Scenarios 4 and 5 have a bit lower expected inventory than scenario 3, which shows the positive influence of the (s, S)-policy instead of the (0, S)-policy on the same amount of items. Between scenarios 4 and 5 themselves is not that much difference. Scenario 4 shows a little bit less expected inventory for every van type.

Scenario 2 shows the influence of implementing the decision flowchart we created. The same list is evaluated as in scenario 1, only now are the items that are classified as floor stock, have a value higher than € 200.00, and are sold less than twice a year left out in this scenario. A decrease in expected inventory value can be seen.

In Section 3.3 we analysed the actual inventory at a certain moment in time, which told us that the average inventory for the catering, retail, and industry vans at the moment lay a little bit above € 8,000. Scenario 1 told us that if Veld would use their current lists the expected value would be much lower than their actual inventory is. Their actual inventory therefore does not really match with what is expected to be in their vans according to the current standard lists. The current standard lists are old, which could be a reason why the actual inventory differs significantly from the expected inventory value. Scenarios 3, 4, and 5 have expected inventory values that came closer to the actual inventory, except for the retail vans.

5.2 Cost evaluation

Int this section will evaluate every scenario based on the holding costs, stockout costs, order costs, and total costs. The results for every scenario are shown in the tables below.

Scenario 1	Holding costs	Stockout costs	Order costs	Total costs
Catering	€ 924.73	€ 20,631.31	€ 30,256.68	€ 51,812.72
Retail	€ 952.20	€ 37,394.80	€ 54,841.03	€ 93,188.03
Industry	€ 645.55	€ 14,887.53	€ 21,833.17	€ 37,366.25
Total	€ 2,522.48	€ 72,913.64	€ 106,930.88	€ 182,367.00

Table 9.1: Costs of scenario 1

Scenario 2	Holding costs	Stockout costs	Order costs	Total costs
Catering	€ 339.13	€ 17,517.80	€ 25,689.52	€ 43,546.45
Retail	€ 508.84	€ 32,397.96	€ 47,512.95	€ 80,419.75
Industry	€ 258.51	€ 12,903.88	€ 18,924.07	€ 32,086.46
Total	€ 1,106.48	€ 62,819.64	€ 92,126.54	€ 156,052.66

Table 9.2: Costs of scenario 2

Scenario 3	Holding costs	Stockout costs	Order costs	Total costs
Catering	€ 1,917.08	€ 1,660.56	€ 3,720.32	€ 7,297.96
Retail	€ 3,370.69	€ 2,417.50	€ 2,983.99	€ 8,772.18
Industry	€ 1,635.27	€ 1,149.33	€ 1,731.16	€ 4,515.76
Total	€ 6,923.04	€ 5,227.39	€ 8,435.47	€ 20,585.90

Table 9.3: Costs of scenario 3

Scenario 4	Holding costs	Stockout costs	Order costs	Total costs
Catering	€ 1,746.47	€ 250.48	€ 2,204.15	€ 4,201.10
Retail	€ 3,182.69	€ 473.91	€ 4,170.30	€ 7,826.90
Industry	€ 1,708.46	€ 114.93	€ 1,011.41	€ 2,834.80
Total	€ 6,637.62	€ 839.32	€ 7,385.86	€ 14,862.80

Table 9.4: Costs of scenario 4

Scenario 5	Holding costs	Stockout costs	Order costs	Total costs
Catering	€ 1,751.57	€ 165.44	€ 2,204,15	€ 4,121.17
Retail	€ 3,204.26	€ 290.97	€ 4,170,30	€ 7,665.53
Industry	€ 1,722.16	€ 97.36	€ 1,011,41	€ 2,830.94
Total	€ 6,678.00	€ 553.77	€ 7,385,86	€ 14,617.63

Table 9.5: Costs of scenario 5

In Section 5.1 we saw that our new proposed solutions lead to a higher expected inventory value than there currently is. In this section it is made clear that the costs of each list can be reduced.

Starting with the holding costs, which is the only cost where the list of scenario 1 performs better than the lists of the other scenarios, except for scenario 2 which is the same list as scenario 1 but now our decision flowchart as filtered out items. As with section 5.1, the expected inventory of the current lists is lower, which means that the holding costs will also be lower for that lists. Scenarios 3, 4, and 5 have the same items on the list, but scenario 3, with an (0, S)-policy, shows a higher expected inventory value than the other two scenarios. This shows that an (s, S)-policy will lead to lower holding costs.

The stockout costs have also been reduced. In scenarios 1 and 2, a stockout would happen in 20% of the reorder cycles, following the current policy of Veld. This led to a high number of possible stockouts, and thus also high stockout costs. In scenario 3 it was still 20% of the reorder cycles with a stockout, but now the number of cycles has reduced which means there are fewer possible stockouts. In scenarios 4 and 5 a safety stock was introduced, based on a CSL, which acts as a buffer to stockouts. Scenario 5 is the scenario with service levels based on the ABC classification, which has lower stockout costs than scenario 4, where we set the service level at 98%. This shows that our classification has a positive influence on the stockout costs.

The last costs we evaluate are the order costs. Remember that in scenarios 1 and 2 an item has a lot of cycles and is thus ordered often. That is why the order costs for this scenario are so high. In scenario 3 we already reduced this. Main reason for this is that the order-up-to-levels of the items are higher, which decreases the number of order cycles of an item in a year. Scenarios 4 and 5 have the same order cost for every type, but what is interesting to see is that the order costs for the catering and the industry have decreased in these scenarios compared to scenario 3, and the order costs for the retail vans have increased. The reason why scenarios 4 and 5 have the same order cost is because the order cost depends on the number of order cycles, which depends on the order quantity and the demand for an item, which are the same for these scenarios.

The total costs show us that an (s, S)-policy decreases the costs the most. Scenarios 4 and 5 evaluate this policy. Scenario 5 has the least total cost, meaning the CSL from our ABC classification has a positive influence and decreases the total cost more than a general CSL of 98%.

5.3 Varying the CSL

In our main research question we said that we would want to maintain our service level and decrease the costs of the inventory. In this section we will see what will happen if we improve our service level. In previous scenarios we already evaluated certain lists based on a general CSL of 98% and a CSL based on our ABC classification. To show the impact of the service level we evaluate the (s, S)-policy on different CSLs. The total cost of the standard list for each CSL is shown in Figure 5.1

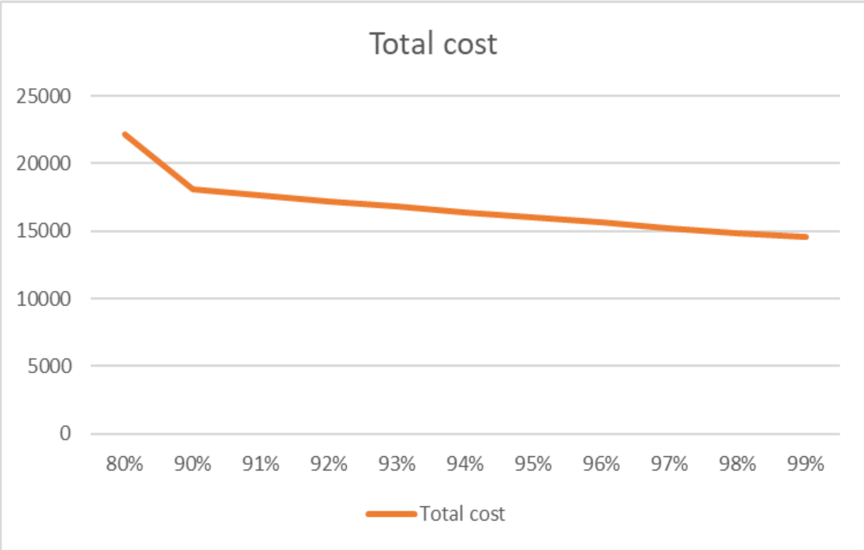

Figure 5.1: The total cost over varying CSLs

This figure shows a pattern, where we can see that our total cost decreases by increasing the CSL. At a CSL of 80%, we obtain a total cost € 22,190.15, meaning that the (s, S)-policy with a CSL of 80% leads to higher total costs than the (0, S)-policy from scenario 3. which is also a scenario with 80% CSL.

This figure also shows that a CSL of 99% gets us the least total cost, which is € 14,597.22. That is less than using the CSL from our ABC classification, as we did in scenario 5. On the other hand, the expected inventory value is higher at a CSL of 99%. The total expected inventory value of our ABC classification is € 33,390.02, while it is € 33,958.03 when using a CSL of 99%. Figure 5.2 shows what the total expected inventory value is for each CSL.

Figure 5.2: Expected inventory value over varying CSLs

5.4 Outcome conclusion

There are a couple of main points that we retrieve from the evaluation of our methodology. We evaluated the lists based on the expected inventory value first, where we saw that the current lists have lower expected inventory value than the proposed solutions. Scenarios 4 and 5 which provide a solution with an (s, S)-policy, have a lower expected inventory value than scenario 3, the scenario which provides a solution using the (0, S)-policy of Veld. These three scenarios all had the same number of items on a list which shows that the (s, S)-policy is a more ideal policy speaking in terms of expected inventory value. The most probable reason for the higher expected inventory value in scenarios 3, 4, and 5 compared to the first two scenarios is the fact that the order-up-to-levels of the first two scenarios are much lower than the levels of the proposed solutions.

Next, we evaluated each list on different costs, where the biggest decrease in cost was found at the stockout costs. Scenarios 1, 2, and 3 do not have safety stocks which leads to an increased chance of stockout and thus also to higher stockout costs. In the policy of Veld, we assume that an order consists of five different items. An order is done every time one of the five runs out of stock. That means that in 20% of the cycles a stockout occurs. In scenarios 4 and 5 safety stock was introduced, reducing the chance of stockouts. Scenario 4 evaluated the costs based on a 98% CSL for every item, which led to higher stockout costs and holding costs compared to scenario 5, where we introduced service levels based on an ABC classification.

The higher order-up-to-levels make sure that an item is less frequently ordered. The fact that an item is less frequently ordered leads to less order costs. Again, using an (s, S)-policy would yield fewer order costs compared to the (0, S)-policy of Veld. In the latter policy, more order cycles occur, since not every item is ordered at the reorder point zero.

The holding costs for every proposed solution list have increased, but this is directly related to higher expected inventory levels for these three scenarios. The increase in holding costs is not that dramatic, compared to the decrease in stockout and order costs.

In scenario 5, we implemented the service levels obtained from the ABC classification. This showed to decrease the total costs more than a general service level of 98% as Veld proposed. In Section 5.3 we saw that a general CSL of 99% decreased the total cost even more, but this came with an increase in expected inventory value.

In our decision flowchart we decided to create a special list, where items would end up if they were sold less than a certain amount of times in a whole year. These items were neglected in all scenarios except scenario 1. For every type, the number of items ending up on this special list would be around 120. A lot of items will not be necessary to be on the standard list but there could be some items that should be on the list. If we did not filter out these items, they would increase the expected inventory value of a list and the holding costs. On the other hand, the inventory level of these items would not be that high since they were not sold that much. The stockout costs and the order cost would not change. The items were not sold meaning that they cannot run out of stock and they do not need to be ordered.

Chapter 6: Conclusion

The motivation for our research was that Veld experienced an increase in its inventory costs. This research focussed on the inventory of service vans of Veld, where we tried to create new inventory lists based on Veld´s current review policy and another inventory review policy, which is the (s, S)-policy. Starting with a decision flowchart, where items will be selected to be in a service van or not. For the (s, S)-policy we did an ABC classification, where we determined a general CSL for each category. The dashboard was then created as an end product for Veld, where we use VBA to generate new standard lists for three types of vans. This dashboard can generate lists based on two policies. One where Veld can continue to use the (0, S)-policy they currently use and one where a proposed (s, S)-policy is generated. In this chapter we will discuss the main findings of this research, some limitations, the recommendations of this research, and we will answer our main research question:

How can we improve the inventory levels of the items within the service vans of the company where we want to decrease the cost but still maintain the service level?

6.1 Main findings

The main findings start with the first step that we took to an improvement and that is the decision flowchart. Although based on logic by the company's wishes, we could reduce the number of unique items in a van by 75% for the catering, going from 380 unique items to 94, 61% for the retail, going from 357 to 139, and 68% for the industry, going from 280 to 90 items. This does not directly say something about how full a van will be since the total quantity of the newer lists can be more than the current list. One could say that the mechanic has less variety of items and thus has a better overview of the items in his van. Plus the items in his vans are, based on the sales data, the items that they use the most.

Using VBA we created a dashboard that generates these inventory lists and evaluates the expected costs of implementing these lists in practice. Evaluation of these lists tells us that using the current (0, S)-policy of Veld generates more costs than an (s, S)-policy as we propose. The stockout costs and the order costs have largely been reduced in the latter policy, which also led to a decrease in the total cost.

Part of our main research question was to maintain the service level. We have shown that even by increasing the service level we could decrease the costs. We found that Veld currently has an 80% CSL, where currently the costs of the standard list are € 182,367.00. In our solution we have shown that in a (0, S)-policy and an (s, S)-policy we could maintain this service level but decrease the total costs by around 89%. In Section 5.3 the total costs decreased even more when increasing the CSL in an (s, S)-policy.

So we have decreased the variety of items in a van, which gives the mechanic a better overview of what he has in his van, we have created a tool giving the company something that creates new standard lists, based on a structured inventory policy, and lastly we have met the goal of this research was to decrease the cost of the inventory but maintain the service level. A side issue is that the inventory values that not increase that much.

6.2 Recommendations

Based on our research we would recommend Veld to implement an (s, S)-policy. The (s, S)-policy has shown to be the policy with the least total costs. The current standard lists have a total cost of € 182,367.00, which can be reduced to € 14,597.22. The latter number is the total costs if Veld uses an (s, S)-policy and a CSL of 99%. That is a decrease in costs of around 92%. We would recommend not using a general CSL of 99%, but the CSL of our ABC classification since the expected inventory value with these service levels is lower than with a general CSL of 99% and the total cost only increases by € 120.41.

One of the main reasons for this decrease in both policies is the use of our decision flowchart. Using this flowchart makes sure only the most essential items end up on the standard list since these lists are generated based on the sales of the service vans. Another reason why the costs have decreased is because the reorder points and order-up-to-levels on the current list of Veld are way lower than in our proposed solution. Therefore the old list had much higher order costs. The new list with higher levels led to much less order costs, but the holding costs did not increase that much since we filtered out a lot of products. That is why a (0, S)-policy could also work for Veld, if the company is not able to implement the (s, S)-policy in its system. Our solution for the (0, S)-policy, where the service level of 80% is maintained, also leads to a decrease in total costs. The total costs will then reduce by 88% to a total of € 22,190.15.

Another reason why we would recommend an (s, S)-policy is for the decrease in stockout costs. Veld has a CSL of 80% which means that an item runs out of stock in 20% of the order cycles, which leads to high stockout costs. That can be seen when comparing scenario 3 to scenarios 4 and 5 in Section 5.2. In scenario 3, where we use the (0, S)-policy with a CSL of 80% we have a total stockout cost of € 5,227.39, while in scenario 5 it is only € 553.77.

Another recommendation would be to not take over the list completely without looking. The dashboard should be seen as a tool to identify the items that should end up on the standard list. The calculations for the reorder points and the order-up-to-levels give the company an idea of what would be ideal. An extremely high order-up-to-level can occur when the holding costs are much lower than the ordering costs. The formula for the EOQ will then determine a high order quantity, which will lead to a high order-up-to-level. If the standard deviation of an item is very high, an extremely high reorder point can occur. People with experience in the service should review the list and identify probable issues with the list. Examples of this are the extremely high order-up-to-levels or reorder points, items from which they know that they will not sell, or items that are missing on the lists. The latter could be debatable since according to the decision flowchart an item does not end up on the list if it is not sold for a whole year. But an experienced service mechanic can think otherwise. Therefore this list can be seen as a right starting point for generating new standard lists.

6.3 Limitations and further research

We had to make some assumptions during this research, to simplify this research. Making these assumptions is one of the limitations of this research, which also gives room for possible further research. We will discuss these topics in this section.

Most of the costs are assumed. There is a fixed cost for ordering products, but there are some extra costs for delivery in Belgium and larger packages. Since we did not take into account the location and the size of a package we did not take into account the possible extra costs associated with this. We also assumed that Veld orders five different items at the same, while in reality they might order more or fewer items. The stockout cost is an assumption as well. We assumed a fixed penalty cost, while in reality Veld does not have to deal with these penalty costs. The holding costs were also an assumption set to 20% of the product value per year.

A simplification we did was using the CSL as an indicator, while the job fill rate might be more suitable. We did not have data available on the fulfilment of a single job, which is used in the formulas for the fill rate. Besides that, the computations for the fill rate are more complex, so the choice for the CSL was one made for simplicity reasons as well. The assumption of one unit short within a stockout instead of multiple is another simplification of the problem. It is not known how many units of an item are needed for a job, for which we assumed the shortage of only one unit.

There is also room for further research when using a CSL as an indicator. We have now determined an average CSL for each category, while it could be more optimal to set a CSL for every individual item. During this research, the choice was once made to write the VBA code for an average CSL. We could change the code to set an individual CSL for every item, but we decided to not do this for this research. If we did, we could achieve fewer costs than a general CSL of 99%, which is now seen as the solution with the least amount of costs.

The assumption of the demand following a normal distribution is a limitation as well. There is no analysis on what distribution a product follows so therefore we assumed a normal distribution since this is the most common distribution. Veld also said they do not deal with seasonal items, but with a certain analysis seasonal items could be identified.

Another limitation of this research is regarding the demand. Veld does not know the demand for its products and does not have any forecasted demand. A mechanic does not know which items are needed for a job. For this we assumed that the demand of 2022 is equal to the demand in 2023, which in reality has a very small chance to occur. The consequence is that items that are not sold in a year will not be on an inventory list, while in reality it might be that a year later the demand for that product is very high.

A limitation that could have a large influence on the inventory lists is the size and weight of certain products. A service van can only handle a certain maximum amount of weight. Veld currently does not keep track of this data, so it could not be implemented in the methodology. If Veld can track this data in its ERP system then this could be a parameter to take into account for further research, by putting it in the decision flowchart and by implementing this as a constraint when generating the inventory lists. It would be quite interesting to see these results, especially considering that Veld currently has more than 300 unique items on the inventory lists.

References

- Atakay, B., Onbaşılı, Ö., Özcet, S., Akbulak, İ., Cevher, H. B., Alcaz, H., Gördesli, İ., Paldrak, M., & Staiou, E. (2022). Spare Parts Inventory Management System in a Service Sector Company. Lecture Notes in Mechanical Engineering, 787–799.
- Bijvank, M., Koole, G., & Vis, I. F. A. (2010). Optimising a general repair kit problem with a service constraint. European Journal of Operational Research, 204(1), 36–48.
- Brumelle, S., & Granot, D. (1993). The repair kit problem revisited. Operations Research, 41(5), 994–1006.
- Chopra, S., & Meindl, P. (2004). Supply Chain Management (5th ed.). Boston: Pearson Education Limited.
- Douissa, M. R., & Jabeur, K. (2020). A non-compensatory classification approach for multi-criteria ABC analysis. Soft Computing, 24(13), 9525–9556.
- Flores, B. E., & Clay Whybark, D. (1986). Multiple Criteria ABC Analysis. International Journal of Operations & Production Management, 6(3), 38–46.
- Graves, S. C. (1982). A multiple-item inventory model with a job completion criterion. Management Science, 28(11), 1334–1337.
- Hausman, W. H. (1982). On optimal repair kits under a job completion criterion. Management Science, 28(11), 1350–1351.
- Heeremans, D., & Gelders, L. (1995). Multiple period repair kit problem with a job completion criterion: A case study. European Journal of Operational Research, 81(2), 239–248.
- Heerkens, H., & Van Winden, A. (2021). Solving Managerial Problems Systematically. Routledge.
- Kaabi, H. (2022). Comparative Analysis of MultiCriteria Inventory Classification Models for ABC Analysis. International Journal of Information Technology and Decision Making, 21(5), 1617–1646.
- Khanorkar, Y., & Kane, P. v. (2023). Selective inventory classification using ABC - classification, multi-criteria decision making techniques, and machine learning techniques. Materials Today: Proceedings, 72, 1270–1274.
- Mamer, J. W., & Shogan, A. W. (1987). A constrained capital budgeting problem with applications to repair kit selection. Management Science, 33(6), 800–806.
- Mamer, J. W., & Smith, S. A. (1982). Optimizing field repair kits based on job completion rate. Management Science, 28(11), 1328–1333.
- Mamer, J. W., & Smith, S. A. (1985). Job completion inventory systems: Optimal policies for repair kits and spare machines. Management Science, 31(6), 703–718.
- March, S. T., & Scudder, G. D. (1984). On "optimizing field repair kits based on job completion rate". Management Science, 30(8), 1025–1028.
- Naranjo-Mantilla, M., Llorens Cervera, S., & Tenesaca, G. (2022). ABC Inventory Control System Based on an Office Automation Tool. Communications in Computer and Information Science, 1654 CCIS, 662–666.
- Ng, W. L. (2007). A simple classifier for multiple criteria ABC analysis. European Journal of Operational Research, 177(1), 344–353.
- Prak, D., Saccani, N., Syntetos, A., Teunter, R., & Visintin, F. (2017). The repair kit problem with positive replenishment lead times and fixed ordering costs. European Journal of Operational Research, 261(3), 893–902.
- Ramanathan, R. (2006). ABC inventory classification with multiple-criteria using weighted linear optimization. Computers & Operations Research, 33(3), 695–700.
- Rippe, C., & Kiesmüller, G. P. (2023). The repair kit problem with imperfect advance demand information. European Journal of Operational Research, 304(2), 558–576.

- Saccani, N., Visintin, F., Mansini, R., & Colombo, M. (2017). Improving spare parts management for field services: A model and a case study for the repair kit problem. IMA Journal of Management Mathematics, 28(2), 185–204.
- Silver, E. A., Pyke, D. F., & Thomas, D. J. (2016). Inventory and Production Management in Supply Chains (4th ed.). Springer.
- Smith, S. A., Chambers, J. C., & Shlifer, E. (1980). Optimal inventories based on job completion rate for repairs requiring multiple items. Management Science, 26(8), 849–852.
- Teunter, R. H. (2006). The multiple-job repair kit problem. European Journal of Operational Research, 175(2), 1103–1116.
- Winston, W. L., & Goldberg, J. B. (2004). Operations Research: Applications and Algorithms (4th ed.). Thomson Learning.
- Zhou, P., & Fan, L. (2007). A note on multi-criteria ABC inventory classification using weighted linear optimization. European Journal of Operational Research, 182(3), 1488–1491.

Appendices

Appendix A: VBA codes

Sub Decision flowchart

```vba
Sub Filter()

Dim i As Integer
Dim j As Integer
Dim lastRow As Integer
Dim OutputSheet As Worksheet
Dim DataSheet As Worksheet
Dim SumSheet As Worksheet
Dim headerA As String
Dim headerB As String
Dim headerC As String
Dim headerD As String
Dim Value As Variant
Dim OutputRange As Range
Dim lastRow1 As Long
Dim horecaSum As Double
Dim retailSum As Double
Dim industrieSum As Double
Dim Sum As Double
Dim searchValue As Variant
Dim OutputRange1 As Range
Dim OutputSheet1 As Worksheet
Dim foundValue As Range


Set OutputSheet = ThisWorkbook.Sheets("Output")
Set DataSheet = ThisWorkbook.Sheets("Data")
Set OutputSheet1 = ThisWorkbook.Sheets("ListToBeChecked")
Set Dashboard = ThisWorkbook.Sheets("Dashboard")

OutputSheet.Range("A2:L1000").Clear
OutputSheet1.Range("A2:C1000").Clear

headerA = OutputSheet.Range("A1").Value
headerB = OutputSheet.Range("D1").Value
headerC = OutputSheet.Range("G1").Value
headerD = OutputSheet.Range("J1").Value
lastRow = DataSheet.Cells(Rows.Count, "A").End(xlUp).Row


For i = 2 To lastRow

Value = DataSheet.Cells(i, "A").Value

If DataSheet.Cells(i, "I").Value = headerA Then
Set OutputRange = OutputSheet.Range("A2:A" & OutputSheet.Cells(Rows.Count, "A").End(xlUp).Row)
Set OutputRange1 = OutputSheet1.Range("A2:A" & OutputSheet1.Cells(Rows.Count, "A").End(xlUp).Row)
horecaSum = DataSheet.Cells(i, "O").Value

If DataSheet.Cells(i, "J").Value < Dashboard.Range("N4").Value Then 'Select which value items you want
If Not DataSheet.Cells(i, "B").Value Like "Koudemidd*" Then
 If DataSheet.Cells(i, "N").Value = "No" Then
If horecaSum > Dashboard.Range("N5").Value Then
Set foundValue = OutputRange.Find(What:=Value, LookIn:=xlValues, LookAt:=xlWhole)
If foundValue Is Nothing Then
OutputRange.Cells(OutputRange.Rows.Count + 1, 1).Value = Value
End If
ElseIf 0 > horecaSum < Dashboard.Range("N5").Value Then
Set foundValue = OutputRange1.Find(What:=Value, LookIn:=xlValues, LookAt:=xlWhole)
 If foundValue Is Nothing Then
OutputRange1.Cells(OutputRange1.Rows.Count + 1, 1).Value = Value
End If
End If
End If
End If

ElseIf DataSheet.Cells(i, "I").Value = headerB Then

Set OutputRange = OutputSheet.Range("D2:D" & OutputSheet.Cells(Rows.Count, "D").End(xlUp).Row)
Set OutputRange1 = OutputSheet1.Range("B2:B" & OutputSheet1.Cells(Rows.Count, "B").End(xlUp).Row)
retailSum = DataSheet.Cells(i, "O").Value

If DataSheet.Cells(i, "J").Value < Dashboard.Range("N4").Value Then 'Select which value items you want
If Not DataSheet.Cells(i, "B").Value Like "Koudemidd*" Then
If DataSheet.Cells(i, "N").Value = "No" Then
If retailSum > Dashboard.Range("N5").Value Then
Set foundValue = OutputRange.Find(What:=Value, LookIn:=xlValues, LookAt:=xlWhole)
If foundValue Is Nothing Then
OutputRange.Cells(OutputRange.Rows.Count + 1, 1).Value = Value
End If
ElseIf 0 > retailSum < Dashboard.Range("N5").Value Then
Set foundValue = OutputRange1.Find(What:=Value, LookIn:=xlValues, LookAt:=xlWhole)
 If foundValue Is Nothing Then
OutputRange1.Cells(OutputRange1.Rows.Count + 1, 1).Value = Value
End If
End If
End If
End If

ElseIf DataSheet.Cells(i, "I").Value = headerC Then

Set OutputRange = OutputSheet.Range("G2:G" & OutputSheet.Cells(Rows.Count, "G").End(xlUp).Row)
Set OutputRange1 = OutputSheet1.Range("C2:C" & OutputSheet1.Cells(Rows.Count, "C").End(xlUp).Row)
industrieSum = DataSheet.Cells(i, "O").Value

If DataSheet.Cells(i, "J").Value < Dashboard.Range("N4").Value Then 'Select which value items you want
If Not DataSheet.Cells(i, "B").Value Like "Koudemidd*" Then
If DataSheet.Cells(i, "N").Value = "No" Then
If industrieSum > Dashboard.Range("N5").Value Then
Set foundValue = OutputRange.Find(What:=Value, LookIn:=xlValues, LookAt:=xlWhole)
If foundValue Is Nothing Then
OutputRange.Cells(OutputRange.Rows.Count + 1, 1).Value = Value
End If
ElseIf 0 > industrieSum < Dashboard.Range("N5").Value Then
Set foundValue = OutputRange1.Find(What:=Value, LookIn:=xlValues, LookAt:=xlWhole)
 If foundValue Is Nothing Then
OutputRange1.Cells(OutputRange1.Rows.Count + 1, 1).Value = Value
End If
End If
End If
End If

ElseIf DataSheet.Cells(i, "I").Value = headerD Then

Set OutputRange = OutputSheet.Range("J2:J" & OutputSheet.Cells(Rows.Count, "J").End(xlUp).Row)
Set OutputRange1 = OutputSheet1.Range("D2:D" & OutputSheet1.Cells(Rows.Count, "D").End(xlUp).Row)
 Sum = DataSheet.Cells(i, "O").Value

If DataSheet.Cells(i, "J").Value < Dashboard.Range("N4").Value Then 'Select which value items you want
If Not DataSheet.Cells(i, "B").Value Like "Koudemidd*" Then
If DataSheet.Cells(i, "N").Value = "No" Then
If Sum > Dashboard.Range("N5").Value Then
Set foundValue = OutputRange.Find(What:=Value, LookIn:=xlValues, LookAt:=xlWhole)
If foundValue Is Nothing Then
OutputRange.Cells(OutputRange.Rows.Count + 1, 1).Value = Value
End If
ElseIf 0 > Sum < Dashboard.Range("N5").Value Then
Set foundValue = OutputRange1.Find(What:=Value, LookIn:=xlValues, LookAt:=xlWhole)
 If foundValue Is Nothing Then
OutputRange1.Cells(OutputRange1.Rows.Count + 1, 1).Value = Value
End If
End If
End If
End If

End If
Next i
End Sub
```

Sub to calculate s and S, part 1

```
Sub CalculateList()

Dim i As Integer
Dim j As Integer
Dim k As Integer

Dim OutputSheet As Worksheet
Dim DataSheet As Worksheet
Dim Dashboard As Worksheet
Dim headerA As String
Dim headerB As String
Dim headerC As String
Dim headerD As String
Dim lastRowA As Integer
Dim lastRowB As Integer
Dim lastRowC As Integer
Dim lastRowD As Integer
Dim lastRowE As Integer
Dim Value As String
Dim OutputRangeA As Range
Dim OutputRangeB As Range
Dim OutputRangeC As Range
Dim foundRow As Range
Dim rowNumber As Long
Dim CellValue As Variant
Dim CellValue2 As Variant
Dim EOQ As Variant




Set OutputSheet = ThisWorkbook.Sheets("Output")
Set DataSheet = ThisWorkbook.Sheets("Data")
Set Dashboard = ThisWorkbook.Sheets("Dashboard")
headerA = OutputSheet.Range("A1").Value
headerB = OutputSheet.Range("D1").Value
headerC = OutputSheet.Range("G1").Value
headerD = OutputSheet.Range("J1").Value
lastRowA = OutputSheet.Cells(Rows.Count, "A").End(xlUp).Row
lastRowB = OutputSheet.Cells(Rows.Count, "D").End(xlUp).Row
lastRowC = OutputSheet.Cells(Rows.Count, "g").End(xlUp).Row
lastRowE = OutputSheet.Cells(Rows.Count, "J").End(xlUp).Row
lastRowD = DataSheet.Cells(Rows.Count, "A").End(xlUp).Row

OutputSheet.Range("B2:B1000").Clear
OutputSheet.Range("C2:C1000").Clear
OutputSheet.Range("E2:E1000").Clear
OutputSheet.Range("F2:F1000").Clear
OutputSheet.Range("H2:H1000").Clear
OutputSheet.Range("I2:I1000").Clear
OutputSheet.Range("K2:K1000").Clear
OutputSheet.Range("L2:L1000").Clear

For i = 3 To lastRowA

x = 0
y = 0
Value = OutputSheet.Cells(i, "A").Value

For j = 2 To lastRowD
If DataSheet.Cells(j, "A").Value = Value And DataSheet.Cells(j, "I").Value = "Horeca" Then
x = DataSheet.Cells(j, "J").Value
CellValue = DataSheet.Cells(j, "L").Value
CellValue2 = DataSheet.Cells(j, "M").Value
y = DataSheet.Cells(j, "O").Value
EOQ = DataSheet.Cells(j, "P").Value
Exit For
End If

Next j



If x > Dashboard.Range("L8").Value Then
If y > Dashboard.Range("L12").Value Then
 OutputSheet.Range("B" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("L24").Value * CellValue) + CellValue2, 0)
ElseIf y > Dashboard.Range("L13").Value And y <= Dashboard.Range("M13").Value Then
OutputSheet.Range("B" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("L25").Value * CellValue) + CellValue2, 0)
ElseIf y <= Dashboard.Range("L14").Value Then
OutputSheet.Range("B" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("L26").Value * CellValue) + CellValue2, 0)
End If

ElseIf x > Dashboard.Range("L9").Value And x <= Dashboard.Range("M9").Value Then
If y > Dashboard.Range("L12").Value Then
OutputSheet.Range("B" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("M24").Value * CellValue) + CellValue2, 0)
ElseIf y > Dashboard.Range("L13").Value And y <= Dashboard.Range("M13").Value Then
OutputSheet.Range("B" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("M25").Value * CellValue) + CellValue2, 0)
ElseIf y <= Dashboard.Range("L14").Value Then
OutputSheet.Range("B" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("M26").Value * CellValue) + CellValue2, 0)
End If


ElseIf x <= Dashboard.Range("L10").Value Then
If y > Dashboard.Range("L12").Value Then
OutputSheet.Range("B" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("N24").Value * CellValue) + CellValue2, 0)
ElseIf y > Dashboard.Range("L13").Value And y <= Dashboard.Range("M13").Value Then
OutputSheet.Range("B" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("N25").Value * CellValue) + CellValue2, 0)
ElseIf y <= Dashboard.Range("L14").Value Then
OutputSheet.Range("B" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("N26").Value * CellValue) + CellValue2, 0)
End If


End If
OutputSheet.Range("C" & i).Value = Application.WorksheetFunction.RoundUp(OutputSheet.Range("B" & i).Value + EOQ, 0)
Next i

For i = 3 To lastRowB

x = 0
y = 0
Value = OutputSheet.Cells(i, "D").Value

For j = 2 To lastRowD
If DataSheet.Cells(j, "A").Value = Value And DataSheet.Cells(j, "I").Value = "Retail" Then
x = DataSheet.Cells(j, "J").Value
CellValue = DataSheet.Cells(j, "L").Value
CellValue2 = DataSheet.Cells(j, "M").Value
y = DataSheet.Cells(j, "O").Value
EOQ = DataSheet.Cells(j, "P").Value
Exit For
End If
```

Sub to calculate s and S, part 2

```
Next j



If x > Dashboard.Range("L8").Value Then
If y > Dashboard.Range("L12").Value Then
 OutputSheet.Range("E" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("L24").Value * CellValue) + CellValue2, 0)
ElseIf y > Dashboard.Range("L13").Value And y <= Dashboard.Range("M13").Value Then
OutputSheet.Range("E" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("L25").Value * CellValue) + CellValue2, 0)
ElseIf y <= Dashboard.Range("L14").Value Then
OutputSheet.Range("E" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("L26").Value * CellValue) + CellValue2, 0)
End If

ElseIf x > Dashboard.Range("L9").Value And x <= Dashboard.Range("M9").Value Then
If y > Dashboard.Range("L12").Value Then
OutputSheet.Range("E" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("M24").Value * CellValue) + CellValue2, 0)
ElseIf y > Dashboard.Range("L13").Value And y <= Dashboard.Range("M13").Value Then
OutputSheet.Range("E" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("M25").Value * CellValue) + CellValue2, 0)
ElseIf y <= Dashboard.Range("L14").Value Then
OutputSheet.Range("E" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("M26").Value * CellValue) + CellValue2, 0)
End If


ElseIf x <= Dashboard.Range("L10").Value Then
If y > Dashboard.Range("L12").Value Then
OutputSheet.Range("E" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("N24").Value * CellValue) + CellValue2, 0)
ElseIf y > Dashboard.Range("L13").Value And y <= Dashboard.Range("M13").Value Then
OutputSheet.Range("E" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("N25").Value * CellValue) + CellValue2, 0)
ElseIf y <= Dashboard.Range("L14").Value Then
OutputSheet.Range("E" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("N26").Value * CellValue) + CellValue2, 0)
End If


End If

OutputSheet.Range("F" & i).Value = Application.WorksheetFunction.RoundUp(OutputSheet.Range("E" & i).Value + EOQ, 0)

Next i

For i = 3 To lastRowC

x = 0
y = 0
Value = OutputSheet.Cells(i, "G").Value

For j = 2 To lastRowD
If DataSheet.Cells(j, "A").Value = Value And DataSheet.Cells(j, "I").Value = "Industrie" Then
x = DataSheet.Cells(j, "J").Value
CellValue = DataSheet.Cells(j, "L").Value
CellValue2 = DataSheet.Cells(j, "M").Value
y = DataSheet.Cells(j, "O").Value
EOQ = DataSheet.Cells(j, "P").Value
Exit For
End If

Next j

If x > Dashboard.Range("L8").Value Then
If y > Dashboard.Range("L12").Value Then
 OutputSheet.Range("H" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("L24").Value * CellValue) + CellValue2, 0)
ElseIf y > Dashboard.Range("L13").Value And y <= Dashboard.Range("M13").Value Then
OutputSheet.Range("H" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("L25").Value * CellValue) + CellValue2, 0)
ElseIf y <= Dashboard.Range("L14").Value Then
OutputSheet.Range("H" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("L26").Value * CellValue) + CellValue2, 0)
End If

ElseIf x > Dashboard.Range("L9").Value And x <= Dashboard.Range("M9").Value Then
If y > Dashboard.Range("L12").Value Then
OutputSheet.Range("H" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("M24").Value * CellValue) + CellValue2, 0)
ElseIf y > Dashboard.Range("L13").Value And y <= Dashboard.Range("M13").Value Then
OutputSheet.Range("H" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("M25").Value * CellValue) + CellValue2, 0)
ElseIf y <= Dashboard.Range("L14").Value Then
OutputSheet.Range("H" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("M26").Value * CellValue) + CellValue2, 0)
End If


ElseIf x <= Dashboard.Range("L10").Value Then
If y > Dashboard.Range("L12").Value Then
OutputSheet.Range("H" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("N24").Value * CellValue) + CellValue2, 0)
ElseIf y > Dashboard.Range("L13").Value And y <= Dashboard.Range("M13").Value Then
OutputSheet.Range("H" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("N25").Value * CellValue) + CellValue2, 0)
ElseIf y <= Dashboard.Range("L14").Value Then
OutputSheet.Range("H" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("N26").Value * CellValue) + CellValue2, 0)
End If


End If
OutputSheet.Range("I" & i).Value = Application.WorksheetFunction.RoundUp(OutputSheet.Range("H" & i).Value + EOQ, 0)
Next i

If x > Dashboard.Range("L8").Value Then
If y > Dashboard.Range("L12").Value Then
 OutputSheet.Range("K" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("L24").Value * CellValue) + CellValue2, 0)
ElseIf y > Dashboard.Range("L13").Value And y <= Dashboard.Range("M13").Value Then
OutputSheet.Range("K" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("L25").Value * CellValue) + CellValue2, 0)
ElseIf y <= Dashboard.Range("L14").Value Then
OutputSheet.Range("K" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("L26").Value * CellValue) + CellValue2, 0)
End If

ElseIf x > Dashboard.Range("L9").Value And x <= Dashboard.Range("M9").Value Then
If y > Dashboard.Range("L12").Value Then
OutputSheet.Range("K" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("M24").Value * CellValue) + CellValue2, 0)
ElseIf y > Dashboard.Range("L13").Value And y <= Dashboard.Range("M13").Value Then
OutputSheet.Range("K" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("M25").Value * CellValue) + CellValue2, 0)
ElseIf y <= Dashboard.Range("L14").Value Then
OutputSheet.Range("K" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("M26").Value * CellValue) + CellValue2, 0)
End If


ElseIf x <= Dashboard.Range("L10").Value Then
If y > Dashboard.Range("L12").Value Then
OutputSheet.Range("K" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("N24").Value * CellValue) + CellValue2, 0)
ElseIf y > Dashboard.Range("L13").Value And y <= Dashboard.Range("M13").Value Then
OutputSheet.Range("K" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("N25").Value * CellValue) + CellValue2, 0)
ElseIf y <= Dashboard.Range("L14").Value Then
OutputSheet.Range("K" & i).Value = Application.WorksheetFunction.RoundUp((Dashboard.Range("N26").Value * CellValue) + CellValue2, 0)
End If


End If
OutputSheet.Range("L" & i).Value = Application.WorksheetFunction.RoundUp(OutputSheet.Range("K" & i).Value + EOQ, 0)
Next i

End Sub
```

Sub to calculate the costs

```
Sub ListCosts()

Dim i As Integer
Dim Dashboard As Worksheet
Dim OutputSheet As Worksheet
Dim DataSheet As Worksheet
Dim HorecaCell As Range
Dim RetailCell As Range
Dim IndustrieCell As Range
Dim CombineCell As Range
Dim lastRowA As Integer
Dim lastRowB As Integer
Dim lastRowC As Integer
Dim lastRowD As Integer
Dim multiply As Integer
Dim rowNumber As Long
Dim CellValue As Variant
Dim CellValue2 As Variant
Dim searchValue As String
Dim HorecaHolding As Range
Dim RetailHolding As Range
Dim IndustrieHolding As Range
Dim HorecaStockout As Range
Dim RetailStockout As Range
Dim IndustrieStockout As Range
Dim HorecaOrder As Range
Dim RetailOrder As Range
Dim IndustrieOrder As Range
Dim HoldingCost As Variant
Dim StockoutCost As Variant



Set Dashboard = ThisWorkbook.Sheets("Dashboard")
Set OutputSheet = ThisWorkbook.Sheets("Output")
Set DataSheet = ThisWorkbook.Sheets("Data")
Set HorecaCell = Dashboard.Range("R5")
Set RetailCell = Dashboard.Range("R6")
Set IndustrieCell = Dashboard.Range("R7")
Set CombineCell = Dashboard.Range("R8")
Set HorecaHolding = Dashboard.Range("W5")
Set RetailHolding = Dashboard.Range("W6")
Set IndustrieHolding = Dashboard.Range("W7")
Set HorecaStockout = Dashboard.Range("Z5")
Set RetailStockout = Dashboard.Range("Z6")
Set IndustrieStockout = Dashboard.Range("Z7")
Set HorecaOrder = Dashboard.Range("AC5")
Set RetailOrder = Dashboard.Range("AC6")
Set IndustrieOrder = Dashboard.Range("AC7")


Dashboard.Range("R5").Clear
Dashboard.Range("R6").Clear
Dashboard.Range("R7").Clear
Dashboard.Range("R8").Clear

Dashboard.Range("W5").Clear
Dashboard.Range("W6").Clear
Dashboard.Range("W7").Clear

Dashboard.Range("Z5").Clear
Dashboard.Range("Z6").Clear
Dashboard.Range("Z7").Clear

Dashboard.Range("AC5").Clear
Dashboard.Range("AC6").Clear
Dashboard.Range("AC7").Clear

HorecaCell = 0
RetailCell = 0
IndustrieCell = 0
HorecaHolding = 0
RetailHolding = 0
IndustrieHolding = 0
HorecaStockout = 0
RetailStockout = 0
IndustrieStockout = 0
HorecaOrder = 0
RetailOrder = 0
IndustrieOrder = 0


lastRowA = OutputSheet.Cells(Rows.Count, "A").End(xlUp).Row
lastRowB = OutputSheet.Cells(Rows.Count, "D").End(xlUp).Row
lastRowC = OutputSheet.Cells(Rows.Count, "G").End(xlUp).Row
lastRowD = OutputSheet.Cells(Rows.Count, "J").End(xlUp).Row


For i = 3 To lastRowA
searchValue = OutputSheet.Cells(i, "A").Value
multiply = OutputSheet.Cells(i, "C").Value
rowNumber = Application.WorksheetFunction.Match(searchValue, DataSheet.Columns("A"), 0)
CellValue = Application.WorksheetFunction.Index(DataSheet.Columns("J"), rowNumber) * (((multiply - OutputSheet.Cells(i, "B").Value) / 2) + OutputSheet.Cells(i, "B").Value)
HoldingCost = Application.WorksheetFunction.Index(DataSheet.Columns("R"), rowNumber)
StockoutCost = Application.WorksheetFunction.Index(DataSheet.Columns("S"), rowNumber)
CellValue2 = HoldingCost * (((OutputSheet.Cells(i, "C").Value - OutputSheet.Cells(i, "B").Value) / 2) + OutputSheet.Cells(i, "B").Value)

HorecaCell = HorecaCell + CellValue
HorecaHolding = HorecaHolding + CellValue2
HorecaStockout = HorecaStockout + StockoutCost
HorecaOrder = HorecaOrder + Application.WorksheetFunction.Index(DataSheet.Columns("T"), rowNumber)
Next i


For i = 3 To lastRowB
searchValue = OutputSheet.Cells(i, "D").Value
multiply = OutputSheet.Cells(i, "F").Value
rowNumber = Application.WorksheetFunction.Match(searchValue, DataSheet.Columns("A"), 0)
CellValue = Application.WorksheetFunction.Index(DataSheet.Columns("J"), rowNumber) * (((multiply - OutputSheet.Cells(i, "E").Value) / 2) + OutputSheet.Cells(i, "E").Value)
HoldingCost = Application.WorksheetFunction.Index(DataSheet.Columns("R"), rowNumber)
AverageInv = (((OutputSheet.Cells(i, "F").Value - OutputSheet.Cells(i, "E").Value) / 2) + OutputSheet.Cells(i, "F").Value)
CellValue2 = HoldingCost * (((OutputSheet.Cells(i, "F").Value - OutputSheet.Cells(i, "E").Value) / 2) + OutputSheet.Cells(i, "E").Value)


RetailCell = RetailCell + CellValue
RetailHolding = RetailHolding + CellValue2
RetailStockout = RetailStockout + Application.WorksheetFunction.Index(DataSheet.Columns("S"), rowNumber)
RetailOrder = RetailOrder + Application.WorksheetFunction.Index(DataSheet.Columns("T"), rowNumber)

Next i

For i = 3 To lastRowC
searchValue = OutputSheet.Cells(i, "G").Value
multiply = OutputSheet.Cells(i, "I").Value
rowNumber = Application.WorksheetFunction.Match(searchValue, DataSheet.Columns("A"), 0)
CellValue = Application.WorksheetFunction.Index(DataSheet.Columns("J"), rowNumber) * (((multiply - OutputSheet.Cells(i, "H").Value) / 2) + OutputSheet.Cells(i, "H").Value)
HoldingCost = Application.WorksheetFunction.Index(DataSheet.Columns("R"), rowNumber)
CellValue2 = HoldingCost * (((OutputSheet.Cells(i, "I").Value - OutputSheet.Cells(i, "H").Value) / 2) + OutputSheet.Cells(i, "H").Value)


IndustrieCell = IndustrieCell + CellValue
IndustrieHolding = IndustrieHolding + CellValue2
IndustrieStockout = IndustrieStockout + Application.WorksheetFunction.Index(DataSheet.Columns("S"), rowNumber)
IndustrieOrder = IndustrieOrder + Application.WorksheetFunction.Index(DataSheet.Columns("T"), rowNumber)
Next i

End Sub
```