Scarce data inventory control

Indrek Pärna

Faculty of Sciences, VU University Amsterdam
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands

January 5, 2017

Preface

In order to complete the master study of Business Analytics at VU University Amsterdam, students are expected to complete an internship at a company and write a thesis on a research topic. The subject of this thesis is the analysis of different replenishment models in the setting where a limited amount of sales data is available, and traditional methods fail. Additionally, the topic of demand distribution stabilization is examined. We investigated the time period after which enough data would be available to make more stable demand forecasts, and when it would be possible to switch to traditional replenishment methods. The internship was done at Slimstock B.V., a company specializing in supply chain optimization. I would like to thank everybody who participated in this project and had a share in its successful completion. Many thanks to Dr. Bartek Knapik (supervisor), Prof. Ger Koole (second reader), Bart van Gessel (Slimstock) and Daan Majoor (Slimstock). Last, but certainly not least, I would like to thank my family and friends for their support.

Amsterdam, December 2016

Summary

Abstract This paper investigates replenishment process models in the so-called (s, Q) setting and compares performance of different policies found in the literature. The analysis is done in a restricted setting where limited amount of data is available, that is, when a new item has been introduced to the assortment. Additionally, we investigate the stabilization of the demand distribution in time.

Slimstock The current thesis was written as part of an internship carried out at Slimstock B.V.. The company is a market leader in inventory optimization in the Netherlands. Slimstock develops forecasting, demand planning and inventory optimization software, and additionally provides consultancy services linked to their software. In order to improve their software, Slimstock wanted to know more about the early stages of product demand; more specifically, what is the best approach when little data is available, and also, when the information about demand stabilizes.

Data Slimstock delivered an anonymized database of sales transactions of an unknown B2B company in the Netherlands. The company has a web shop, physical stores, a central distribution center and sells climate control systems. All items in the data set had a maximum of seven months of sales history. After pre-processing and filtering the dataset contained 456 items and 12743 corresponding sales transactions.

Model In our comparison we used order, holding and lost sales costs to measure performance of each replenishment policy. Daily demand was modeled as comprising three components: items bought by first time buyers, items bought by returning buyers and returned goods. To model daily demand components, we used Poisson and negative binomial distributions. We

used Bayesian inference to estimate the distribution of the parameters of the probability distributions based on the historical data.

Simulation In total, we compared six different replenishment policies found in literature. For each item, replenishment policy and demand distribution type combination, we ran a simulation over the history of the item sales data. For each of the days we created a model based on the historical data, simulated the future demand using the model and then used the simulated data in the replenishment model to make a decision about how large replenishment order should be placed. At the end of the simulation, we were able to calculate cost statistics, and the point where the demand distribution stabilized.

Contents

Chapter 1

Introduction

1.1 Setting the stage

Structure of the thesis In order to make reading the thesis easier, we will now lay out the structure of the paper. In the current chapter, we will talk about general setting where replenishment models are used. We will also state our research problems and hint about what techniques we used to come to a solution. In Chapter 2, we introduce basic terminology and concepts. We will also talk about notation that will be used throughout the paper. Most importantly, we will cover different policies found in literature. We will later include these policies in the comparison. In Chapter 3, we will describe the data that was used in the research. Next, we will explain in detail how we modeled demand, and how the model was used in the simulation. We will also explain how we analyzed the question of demand distribution stabilization. In Chapter 4, we will show the results of the research and provide an analysis of the winning replenishment policy. We will conclude with Chapter 5, where we talk about some problems with the current research and possible future improvements.

About the company Current thesis was written as part of an internship carried out at Slimstock B.V.. The company was established in 1993 and has become a market leader in inventory optimization, with more than 650 customers globally. Slimstock develops forecasting, demand planning and inventory optimization software and also provides consultancy services related to their software. In order to improve their software, Slimstock wanted to know more about the early stages of product demand, more specifically what the best approach is when little data is available, and also when the information about demand stabilizes. This was the starting point for research.

Basics In every business that deals with goods, it is important to keep sufficient stock on hand to support the business. In simple inventory models, the stock on hand is analytically split into base stock and safety stock. Base stock is the expected demand. Safety stock is the amount of inventory kept on hand to protect against uncertainties in customers' demand and supply of items [7].

Forecasting When goods are ordered to replenish the warehouse or store, the ordered goods do not arrive immediately. Due to these lead times from the point of ordering to the delivery of goods, forecasts are used to plan ahead. During the lead time, customers will still be purchasing goods; the quantity that is ordered by the customers can be modeled as a random variable. In order to have a good forecast, it is important to estimate the probability distribution of demand in lead time. Using the distribution, it is possible to obtain more insight into expected demand as well as the uncertainty of it. The lack of a good forecast can lead to overstock or out-of-stock situation.

Costs There are several types of costs associated with keeping items in stock. Goods have to be purchased, which results in ordering costs. There are also costs related to keeping the stock in the warehouse or store, the so-called holding costs. From the perspective of keeping these costs low, it makes sense to have as little inventory as possible. On the other hand, if a customer wants to buy an item and it is not available, this results in lost sales and possible negative reputation. Lost sales can easily be calculated as potential profit that could have been made, whereas negative reputation is difficult to quantify. From the perspective of minimizing lost sales, it makes sense to have as large an inventory as possible. The optimal solution lies somewhere in between these two perspectives.

1.2 Goals of the research

Policy Given a demand forecast/distribution, replenishment policy determines how the orders will be placed. One common policy type is the so-called

(s, S) policy that orders enough to reach inventory level S as soon as inventory level s is reached. Another common type of policy is usually referred to as (s, Q) policy, which orders a quantity of Q as soon as inventory level s is reached. In the current study, we focus on different variations of (s, Q) policies. We look at daily data and also place replenishment orders according to the policy at the daily rate. The exact policies will be covered later in Section 3.3. We want to find the replenishment policy which best allows to lower costs and reduce out-of-stock situations.

Scarce data As demand forecasting is such an important topic in business, there is research available about it [12, 20]. As far as we know, all the research so far focuses on the situation where sufficient data is available to have a good estimate of the demand distribution. Our research is focused on the situation where a limited amount of data is available about the demand: the item has just been introduced to the assortment. This makes the estimation process of the demand distribution, and thus forecasting procedure as well, more complicated.

Stabilization As we are dealing with a situation where we have very little data available, it makes sense that in the beginning, the forecast might have considerable uncertainty to it. The best policy to use in such a situation might not be the same as when enough data is available to estimate demand distribution more accurately. Therefore, it is also interesting to investigate when do we have sufficient data to have more stable forecasts. Stability of a distribution of demand is an ambiguous term, so, we have defined a metric to measure when the stabilization takes place.

Research problem The aim of the research is to compare replenishment policies that could be used in the setting where a limited amount of data is available. We will compare the policies with regards to their order, holding and lost sales costs. We will also provide an estimate of the time period after which the demand distribution stabilizes.

Simulation process In order to find answers to our research problem, we built software that simulated using replenishment policies. This meant that for each of the days we created a model based on the historical data, simulated the future demand using the model, and then used the simulated data in the replenishment model to make a decision about how large a replenishment order should be placed. At the end of the simulation, we were able to calculate cost statistics and when the demand distribution stabilized. Later, we aggregated costs across policies, and the stabilization metrics across demand distribution type. The detailed structure of the model is laid out in Sections 3.3 and 3.4.

Chapter 2

Mathematical foundation of the problem

Chapter flow In this chapter, we will cover general concepts of mathematical modeling, giving the reader an idea of how probability theory is used to solve problems. We will continue by introducing the the types of distribution functions that will be used in this research. We will also explain why we opted to use a Bayesian approach in our modeling process. In Section 2.2 we will give detailed explanations of replenishment models found in literature. Additional important topics which are not covered here will be mentioned in Section 2.3.

2.1 General concepts

Research problem As already stated, the aim of the research is to compare replenishment policies that could be used in a setting where a limited amount of data is available. We also want to provide an estimate of the time period after which the demand distribution stabilizes. We will use mathematical modeling in order to describe the demand process and analyze the problem.

Modeling and probability Inherently, the process of customers buying and returning products depends on so many factors that are beyond our control that it is virtually a random process. Yet, the fact that the process is random does not mean that the process has no structure at all. That structure in probability theory is described by probability distribution, which maps the possible outcomes to the probability of the outcome actually happening. There are many probability distributions that have been investigated in the past and have well-known properties. For an overview, the reader should refer to any textbook on probability theory. For instance, Walpole et al. have covered introductory topics on statistics and probability theory [10]. Given the data, a modeler thinks of the data generating process, uses the probability distributions as building blocks, and builds a mathematical model. That process is called modeling. The mathematical model can then be used to analyze the behavior of the system, in our case customers' demand. We will explain our models in detail in Section 3.3.

Demand distributions In modeling the demand, one of the most important decisions is the choice of underlying probability distribution. Poisson and negative binomial distributions are the most common choices for discrete demand as they both support only non-negative integer values. Negative binomial distribution could be viewed as an extension to Poisson distribution with the variance not being equal to the mean. Given the parameter λ, the probability that a Poisson distributed random variable takes the value k is calculated with (2.1). Given the parameters r and p, the probability that a negative binomial distributed random variable takes the value k is calculated with (2.2). Normal distribution could be used if demand per period is large because the modeled demand should be non-negative [7, 18]. This means that normal distribution could be used when the location of the distribution is high enough and the scale is small enough, so that the negative values have virtually zero probability. However, in the current research we will not focus on normal distribution.

$$\Pr(X_{\text{Poisson}} = k) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{2.1}$$

$$\Pr(X_{\text{Negative binomial}} = k) = \binom{k+r-1}{k} \cdot (1-p)^r p^k \tag{2.2}$$

Bayesian inference The selection of demand distribution function introduces a new problem: the choice of parameters of the selected distribution. One option is to use the so-called point estimates of parameters to forecast future demand. Another option is to use Bayesian inference to estimate the distribution of the parameters themselves. This approach gives a more multidimensional result, and although computationally more intensive, it allows one to have insight into the variation of demand related to the uncertainty of the parameter values. For that insight, we opted to go for the Bayesian approach. Bayesian inference requires one to have a prior belief in the outcome which gets updated when evidence is taken into account, that is, prior distribution gets updated with data to obtain posterior distribution.

Bayes' theorem Bayes' theorem (2.3) links probabilities of events A and B with the conditional probabilities of events $A \mid B$ and $B \mid A$. For instance, $\Pr(A \mid B)$ denotes the probability of event A, given that event B is true. In our case, we will be interested in the probability distribution of parameter θ, given the data, which can be calculated with (2.4).

$$\Pr(A \mid B) = \frac{\Pr(B \mid A)\ \Pr(A)}{\Pr(B)} \tag{2.3}$$

$$\Pr(\theta \mid \text{data}) = \frac{\Pr(\text{data} \mid \theta)\ \Pr(\theta)}{\Pr(\text{data})} \tag{2.4}$$

Notation As different authors that we refer to have opted for different notations, we have reviewed and unified it in order to be consistent and clear. In the current work, notation, described in Table 2.1, will be used. It is important to explain the difference in the notation between $F_t^{-1}(q)$ and $D_{t,q}$, including the reason why different notations are used. In the current chapter, we review literature and the formulas that use $F_t^{-1}(q)$, which denotes the inverse distribution function also known as the quantile function. The formulas that are used assume that the exact distribution is known. On the other hand, in the current study we use simulations to generate samples of total demand and use sample quantiles as proxies for the theoretical quantiles. We will use $D_{t,q}$ to denote sample quantiles.

2.2 Review of various policies

Different policies We will now continue by introducing different approaches to replenishment policies. It is important to mention at this point that each replenishment policy assumes that demand distribution is known: probability distribution function and its parameters. In principle, any replenishment

Type	Notation	Description
Model	z	Demand component
	R	Length of the review period
	D_t	Demand during t days
	$D_{t,q}$	q quantile of demand distribution during t days
	$F_t^{-1}(q)$	Inverse distribution function of demand within t time units
Data	x	Current stock position
	d_z	Observed data
	y_t	Arriving order on day t
	n	Length of observed data
	L	Lead time
Costs	c_h	Holding costs per item per unit of time
	c_{ls}	Lost sale cost
	c_o	Fixed cost of ordering
Policy dependent	s	Reorder level
	Q	Order quantity
	S	Order-up-to level

Table 2.1: Notation used throughout the paper.

policy must determine a replenishment order quantity at any point in time. The way the order quantity is determined divides policies into different categories. Also, the recalibration of the policy and ordering might occur either after a certain review period or continuously. Bijvank summarizes different possible approaches to replenishment policies as shown in Figure 2.1 [7]. Although different policies seem to calculate different values S and Q, in simple models there is a relationship between S and Q , which simply means that the order size should lead to the order-up-to level, taking into account the current stock position and the previous orders arriving within the lead time (2.5).

$$Q = \max\left\{S - x - \sum_{i=1}^{L-1} y_i, 0\right\} \tag{2.5}$$

The intuition behind the formula is that if you know your target inventory level, then you will place an order that will take you to the target level. In other policies, the relationship might not be as straightforward.

		order moment	
		continuous review	periodic review
order size	fixed	(s, Q)	(R, s, Q)
	variable	(s, S)	(R, s, S)

Figure 2.1: Different types of replenishment policies according to Bijvank [7].

Order-up-to level As seen from equation (2.5), if we know S, then we can easily calculate Q. Koole and others show that, with equation (2.6)[1], it is possible to calculate optimal order-up-to level S [7, 20, 13, 21, 17]. The intuition behind this formula is that the optimal order-up-to level is related to the probability of running out of stock during the lead time. The probability is determined by the relationship of the lost sales costs to the total costs and the probability distribution of demand in lead time. The ratio of the cost of lost sales to the total costs, $\frac{c_{ls}}{c_{ls}+c_h}$, should be equal to the probability of the demand not exceeding the optimal order-up-to level. In other words, S is determined using the quantile function of the demand distribution in the lead time (2.6). If lost sale costs are relatively low compared to the total costs, then S will also be low.

$$S = F_L^{-1}\left(\frac{c_{ls}}{c_{ls} + c_h}\right). \tag{2.6}$$

Restricted policies Naturally, research has been done into the extensions to the classical order-up-to level policies. One example is the so-called restricted model where an order is always at most of size $\bar{q}$. The idea is that the classical order size $S - x - \sum_{i=1}^{L-1} y_i$ is restricted by some value $\bar{q}$.

$$Q = \min\left\{S - x - \sum_{i=1}^{L-1} y_i, \bar{q}\right\} \tag{2.7}$$

[1] In literature also F_{L+1}^{-1} is found. This is due to the ambiguity in defining lead time. The extra day of handling goods might already be included in L or not.

In his work, Morton proposes to restrict the order size Q by the daily demand quantile [17].

$$\bar{q} = F_1^{-1}\left(\frac{c_{ls} - c_o}{c_{ls} - c_o + c_h}\right) \tag{2.8}$$

Johansen and Thorstenson et al. propose a very similar formula [2] [20].

$$\bar{q} = \frac{S}{L} \tag{2.9}$$

In both restricted policies the value of $\bar{q}$ is determined based on the daily demand. In (2.8) the ratio of lost sales costs to the total costs (corrected for fixed order costs) should be equal to the probability of the daily demand not exceeding $\bar{q}$. In (2.9), it is assumed that the optimal-up-to level order will be evenly sold in the lead-time, hence the formula.

Ordering decision Koole, in his work [13], presents another extension to the classical model. For a given order size Q, it is possible to make a decision if it makes sense, in terms of costs, to place the order. According to this decision rule, one should only place an order of size Q if the cost of staying at the current inventory level is higher than the cost of placing the order. That is, if the condition in (2.10) is met, where $C(S)$ is defined in (2.11).

$$C\left(x + Q + \sum_{i=1}^{L-1} y_i\right) < C\left(x + \sum_{i=1}^{L-1} y_i\right) - c_o \tag{2.10}$$

$$C(S) = c_o + c_h\,\mathbb{E}(S - D_L)^+ + c_{ls}\,\mathbb{E}(D_L - S)^+ \tag{2.11}$$

The function $C(S)$ can be understood in the following way: if the order-up-to level exceeds the demand in the lead time, the costs are $c_o + c_h(S - D_L)$, since not everything will be sold, and the surplus will induce additional holding costs. On the other hand, if the demand is higher than the order-up-to level, it will result in lost sales, and the costs are $c_o + c_{ls}(D_L - S)$.

Iterative method Additionally, Koole mentions formulas (2.12), (2.13), where Q and s are calculated with an iterative method until convergence. The method is iterative because s depends on Q and vice versa. Expected lost sales in the lead time function $b(s)$ are calculated with (2.14). Koole

[2] Also here $L + 1$ could be used due to the ambiguity in defining lead time.

derives these formulas by minimizing order, inventory and lost sales costs. Koole writes about this approach in the section about multi-order models, whereas previous models were described as being appropriate for single-order settings.

$$Q = \sqrt{\frac{2\,\mathbb{E}\,(\frac{D_L}{L})(c_o + c_{ls}b(s))}{c_h}} \tag{2.12}$$

$$s = F_L^{-1}\left(1 - \frac{2c_hQ}{2c_{ls}\,\mathbb{E}\,(\frac{D_L}{L}) + c_hQ}\right) \tag{2.13}$$

$$b(s) = \mathbb{E}\,(D_L - s)^+ \tag{2.14}$$

2.3 Additional topics

Unobserved demand One important topic related to demand forecasting is the so-called unobserved demand. If one looks at sales data and specifically the moments when the inventory level reaches zero, one sees that orders could have been made at those moments, if the inventory had been there. For example, imagine a retail store where, if an item is not available, the customer will not make the purchase and will leave the store. If these potential orders are not recorded, then information about demand is lost. According to Bijvank demand estimation, although difficult to obtain, should appropriately account for any unobserved demand when sales data is used. Otherwise, the demand is underestimated [7]. The same topic is also covered by others [9, 12, 16, 21]. Chen has derived heuristics to deal with the computational issues of estimating unobserved lost sales [8]. In the current study, the unobserved demand data are not used due to the fact that in the given data set we only had sales data per date and not the true inventory levels. This meant that we did not know if or when the inventory level reached zero.

Additional research In addition to the aforementioned topics, we will now refer to other related issues. Bijvank et al. have investigated models with varying lead times and derived optimal policies for specific cases using Markov chains and dynamic programming [15]. Perishable inventory problems are covered by Chen [8]. Liyanage et al. show how to use operational statistics to optimize order quantities and compare their methodology to others that use point estimates for model parameters [14]. A book by Graves et al. covers a wide range of topics related to inventory control systems and is a great source for general understanding [19].

Chapter 3

Attacking unknown demand

Chapter flow In this chapter, we will explain how the process of research was carried out. We will introduce the technical tools that we used, the given data and the pre-processing that we applied to it. We will continue by explaining the process of simulation in detail, including the models that we used and the analysis we executed before and after simulations.

3.1 Technical tools

Python During this study, Python programming language [11] was used for the data analysis. Most notably, Python package Pandas [4] was used for pre-processing, data visualization and exporting results to LaTeX. For Bayesian analysis, we used Python MCMC implementation package pymc[6] which implements the Metropolis-Hastings algorithm to estimate posterior distributions[3]. In addition, Eclipse [2], in combination with pydev [5], was used for parallel processing during number crunching as well as for debugging coding issues.

3.2 Data preparation

Given data Our research was based on an anonymized database of sales transactions of a B2B company in the Netherlands. The company sells climate control systems, has a web shop, physical stores, and a central distribution center. All items in the database had less than seven months of sales history and were sold in discrete quantities. The database was split into two tables: master data and transactions data. In the master data table, we had item-specific information: lead time in calendar days, cost price, sales price, holding cost and starting inventory. In the transactions table, we had information about sales transactions and returns per date. Specifically, we knew per date and customer the quantity of a product bought or returned. The transactions database started with the first sales date. As seen in Table 3.1, the initial master data table contained 1,927 items in 11 warehouses. There, CDC stands for central distribution center. An example of master data can be seen in Table 3.3. The transactions table contained 26,005 entries. The split between different transaction types is shown in Table 3.2. An example of transaction data can be seen in Table 3.4. It should be noted that in our analysis, it is assumed that the replenishment orders are placed at the end of day, and that the orders arrive in the morning, after which, additional processing is needed. This means that the goods effectively become available for selling the day after. For that reason, an adjustment of one day is added to the original lead time.

Transaction types In the current study, we decided to split the total demand of the day into three sub-components: first-time buyers, returning buyers and returns. The motivation for doing this was the intuition that the three processes are in reality intrinsically distinct. First-time buyers are related to the process of knowledge about the product spreading in the total population of buyers. Returning buyers relate to more mature product demand when customers already are familiar with the product. Finally, returns are related to the product quality and customer satisfaction. Modeling the three processes separately should give more accurate forecasts.

Filtering In order to obtain a meaningful result from the research, we had to ensure that the quality of our data was appropriate. We will now give a list of checks that we ran, along with the reason why.

- Check that the first transaction of a customer is not a return; otherwise something would be returned that has not been sold yet.
- Check that the first issue is less than the initial position; this would result in negative inventory.
- Check that we have at least 2 lead times of history; an arbitrary threshold to ensure that we have a certain length of data available.

warehouse	#items
CDC_	1124
Shop10_	147
Shop1_	99
Shop2_	112
Shop3_	119
Shop4_	49
Shop5_	35
Shop6_	68
Shop7_	102
Shop8_	58
Shop9_	14
Total	1927

Table 3.1: Number of items per warehouse in the initial data set of master data.

Transaction type	Count
Buying first time	12174
Buying returning	12780
Return	1051
Total	26005

Table 3.2: Number transactions per transaction type in the initial data set of transactions data.

warehouse	code	leadtime_in_days	FirstSalesDate	CostPrice	SalesPrice	HoldingCostPercOfCostPrice	startPosition
CDC_	10602333	7	2015-03-16	23.17	30.1442	20	8
CDC_	10602359	7	2014-08-12	26.46	34.9740	20	1
CDC_	10608542	10	2015-02-23	2.74	3.8242	20	11
CDC_	10608599	7	2015-05-28	9.87	11.7177	20	111
CDC_	10614712	10	2014-08-22	24.01	28.4640	20	1

Table 3.3: Master data example.

- Check that lead time, cost price, holding cost and sales price would be positive; by definition these values should be positive.

- Check that the sales price exceeds the cost price; according to the models that are studied in this paper, items making a loss should not

warehouse	code	issueDate	CustomerID	issueQuantity	isReturn	isFirstTimeCustomer
CDC_	10602333	2015-03-16	7828	2	False	True
CDC_	10602333	2015-03-17	7828	1	False	False
CDC_	10602333	2015-03-23	52755	1	False	True
CDC_	10602333	2015-03-30	52755	2	False	False
CDC_	10602333	2015-04-01	7828	2	False	False

Table 3.4: Transaction data example.

be sold at all.

- Check that the initial stock position would be positive; negative stock does not make sense in the current context.

Resulting data All items that did not meet the requirements described in the previous paragraph were removed from the data set. Table 3.5 shows that we were left with 456 items, and Table 3.6 shows that there were 12,743 corresponding transactions. In Table 3.7, we can see some statistics of the counts of transactions per transaction type in the remaining data. Table 3.8 shows the average time between orders. In Figures 3.1 and 3.2 we have displayed some histograms of sales and returns data. The histograms show the number of orders per order size for a selection of items.

warehouse	#items
CDC_	396
Shop10_	4
Shop1_	12
Shop2_	12
Shop3_	14
Shop4_	4
Shop5_	1
Shop6_	4
Shop7_	7
Shop8_	2
Total	456

Table 3.5: Filtered master data data set: number of items per warehouse.

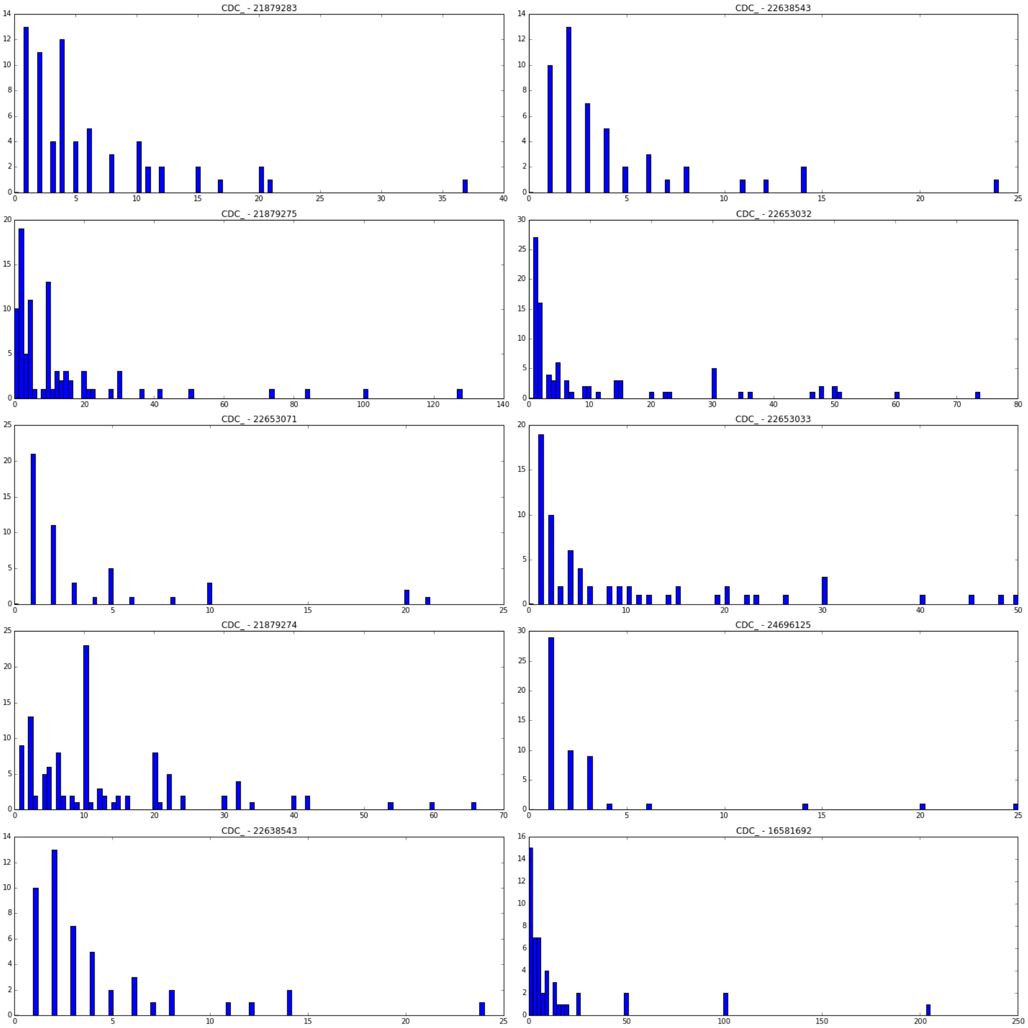


Figure 3.1: Example histograms of sales data: daily sales figures and their counts in the history of an item.

3.3 The model and its updating

Model Our main goal of modeling the demand was to get a probability distribution of the demand within lead time. In order to achieve this, we split the model into sub-components: demand in lead time is a sum of demands in each day and daily demands are distributed according to some assumed parameterized distribution. The so-called hyperparameters of the

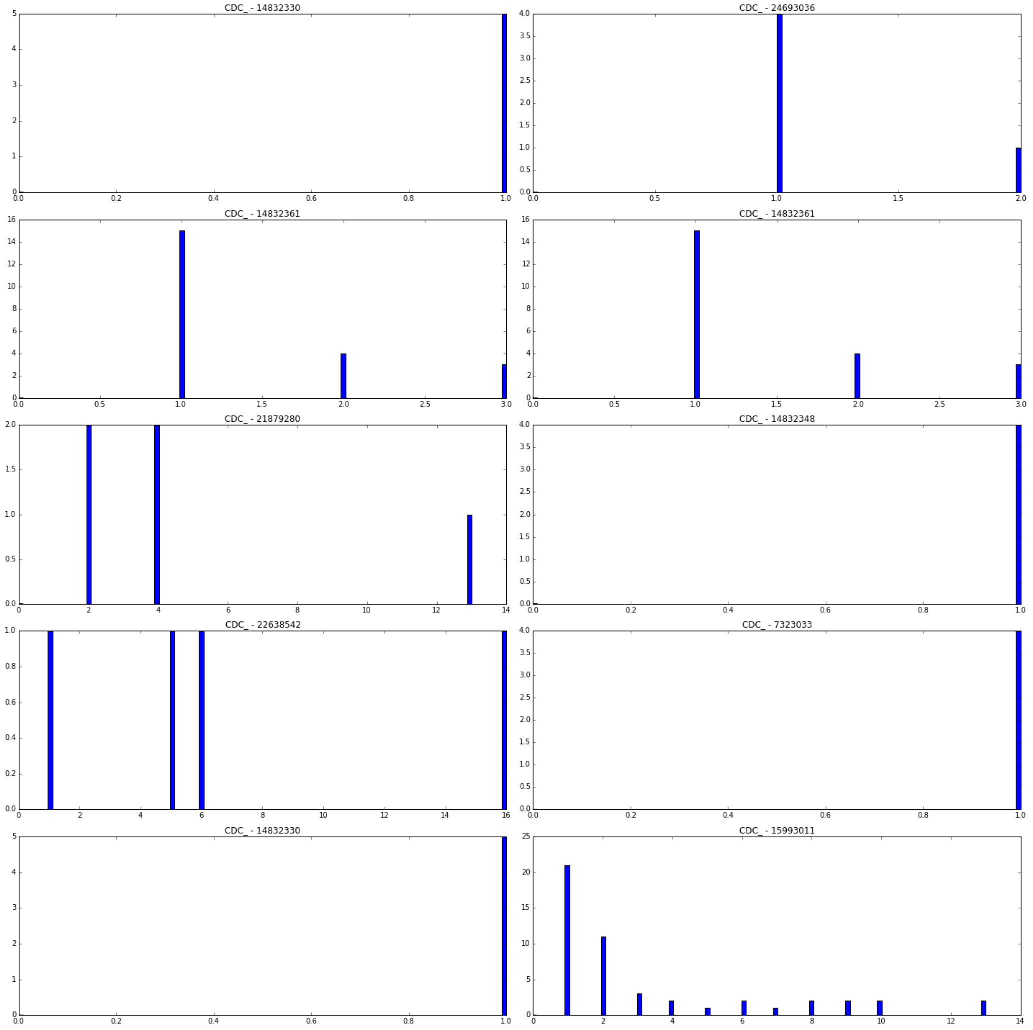


Figure 3.2: Example histograms of returns data. Daily return figures and their counts in the history of an item.

daily demand distribution also have a distribution for which we use our prior belief. Using the observed data, our prior distributions were updated to get posterior distributions of the hyperparameters. We could then draw many realizations from the distributions of hyperparameters and use them as inputs to the daily demand distribution to simulate daily demands. The sum of those daily demands gave us the simulated demands in lead time which

Transaction type	Count
Buying first time	5963
Buying returning	6325
Return	455
Total	12743

Table 3.6: Filtered transactions data set: number of items per transaction type.

	TransactionCountTotal	TransactionCountReturn	TransactionCountBuyingFirstTime	TransactionCountBuyingReturning
mean	27.95	1.00	13.08	13.87
std	153.99	3.64	30.51	123.59
min	3.00	0.00	1.00	0.00
25%	5.00	0.00	4.00	0.00
50%	8.00	0.00	6.00	2.00
75%	18.00	1.00	12.00	6.00
max	3182.00	66.00	521.00	2595.00

Table 3.7: Overview of statistics of transaction counts per transaction type.

	AverageTimeBetweenTransactions
count	456
mean	27 days 08:56:16.782458
std	22 days 13:03:57.422526
min	0 days 02:44:46.702294
25%	10 days 09:04:00.973630
50%	21 days 00:00:00
75%	39 days 11:08:34.285714
max	109 days 08:00:00

Table 3.8: Overview of statistics of average time between transactions.

we could use as a proxy for the true distribution. These simulations were later used by the replenishment policies to make decisions about how to place orders. It is important to mention here that the updating of the model took place with daily frequency; per day we used the historical data to estimate the parameters of the probability distributions of the daily demand.

Demand components Additionally, as already mentioned in Section 3.2, we decided to split demand in each day into three components: first time buyers, buyers who are returning to buy the same item and returns. Therefore, total demand in each day D is calculated with the following formula:

$$D_1 = D_{\text{buying first time}} + D_{\text{buying returning}} - D_{\text{returning}}. \tag{3.1}$$

Poisson demand The first demand distribution that we investigated was Poisson distribution. We used Gamma distribution as prior on the parameter of Poisson distribution. This was convenient as prior and posterior distribution of parameter λ are then in the same family, and we could use an explicit form of posterior distribution [1].

$$z \in \{\text{returning}, \text{buying first time}, \text{buying returning}\} \tag{3.2}$$

$$\lambda_z \sim \Gamma(1,1)$$

$$D_z \sim \text{Poiss}(\lambda_z)$$

$$\lambda_z | d_z \sim \Gamma\left(1 + \sum_{i=1}^{n} d_{z,i}, 1 + n\right)$$

For simulating daily demands, at least 5,000 independent realizations of λ_z were drawn from the posterior distribution. The sample size topic is discussed further in Section 3.5. The realizations were then used as inputs for the Poisson distribution to draw daily demand values. Daily demands were then used to calculate demand within lead time with formula (3.3). As a result, we had an array of simulated total demands in lead time.

$$\begin{aligned} D_L = \sum_{i=0}^{L} [&\text{Poiss}(\lambda_{\text{buying first time}} | d) \\ &+ \text{Poiss}(\lambda_{\text{buying returning}} | d) - \text{Poiss}(\lambda_{\text{returning}} | d)] \end{aligned} \tag{3.3}$$

Negative binomial demand The second demand distribution that we investigated was negative binomial distribution. Due to the fact that we did not want to fix either of the parameters of negative binomial distribution, there was no explicit form for the posterior distribution of the parameters. This meant that we had to resort to an MCMC approach. We used the r and p parametrization, where r denotes the number of failed trials in a sequence of experiments and p denotes the probability of success of each experiment.

In our model, the prior Poisson distribution of r_z was shifted in order to ensure strictly positive values of r.

$$r_z \sim \text{Poiss}(1) + 1$$

$$p_z \sim U(0,1)$$

$$D_z \sim NB(r_z, p_z)$$

In this case, in order to simulate demand in lead time, the whole model was inserted into the MCMC software. The whole model consisted of observed data, priors, daily demand split, daily demand having negative binomial distribution, and summing of the daily demands to get demand in lead time. The array of simulated total demands in lead time was the output of the so-called trace of the MCMC run.

Costs In our simulations of using different replenishment policies, there were three types of costs involved: holding costs, lost sales costs and order costs.

$$\text{total costs} = c_h + c_o + c_{ls} \tag{3.4}$$

where c_{ls} = sales price − cost price, c_o is defined globally and c_h is defined per item and warehouse combination.

Policies In Table 3.9 we present all the policies that were implemented during the research. Here, q is defined by equation (3.5) and $b(s)$ is defined in (2.14). For each of the policies, both Poisson and negative binomial distributions were used when modeling demand. One exception is the random policy which does not use any information from the observed data; thus, modeling was skipped. The random policy is used as a baseline to compare against.

$$q = \frac{c_{ls}}{c_{ls} + c_h} \tag{3.5}$$

Main flow In order to compare different policies, we had to simulate the policy being used. For each combination of item, warehouse, policy and underlying demand distribution type, we ran through the history of the item. For each date in the history of the item, we initialized policy using the history of demand until that point in time and let the policy place an order. The policy then used the simulated demand in lead time to make a decision to

Name	s	Q	Order decision
Random	$U(5,30)$	$U(5,20)$	None
Fixed quantile	$D_{L,0.95}$	$S-x-\sum_{i=1}^{L-1} y_i$	See (2.10)
Dynamic quantile	$D_{L,q}$	$S-x-\sum_{i=1}^{L-1} y_i$	See (2.10)
Dynamic quantile restricted	$D_{L,q}$	$\min\{S-x-\sum_{i=1}^{L-1} y_i, \frac{S}{L}\}$	See (2.10)
Dynamic quantile Q=S	$D_{L,q}$	S	See (2.10)
Dynamic quantile Q=halfS	$D_{L,q}$	$\frac{S}{2}$	See (2.10)
Multi order	$D_{L,\left(1-\frac{2c_hQ}{2c_{ls}\mathbb{E}(\frac{D_L}{L})+c_hQ}\right)}$	$\sqrt{\frac{2\mathbb{E}(\frac{D_L}{L})(c_o+c_{ls}b(s))}{c_h}}$	None

Table 3.9: Implemented policies.

order. We recorded the decision and kept track of the orders in transit. Once the loop over dates was completed, we calculated lost sales, average inventory for holding costs and number of placed orders for order costs. Figure 3.3 shows a flow diagram of the process of simulation. In Figure 3.4 we show an example of a simulation run.

3.4 Demand distribution stabilization

Necessity As already discussed in Chapter 2, the current study focuses on the setting where a very limited amount of data is available. In this setting, initially inaccurate and fluctuating forecasts must be expected. The reason for such inaccuracy is the lack of information. The reason for fluctuations is that each data point will have a strong impact on the demand model. It is thus interesting to investigate at which point we could say that enough data is available to have more stable forecasts.

Metric As part of the simulation process, per day, using the demand available until that moment in time, we estimated the distribution of the demand within lead time. We then stored the 0.05, 0.5 and 0.95 quantiles of the demand distribution. Later, using this information, we were looking for the first day during the simulation run when for all the following days the median of the distribution lay in between the 0.05 and 0.95 quantiles of that partic-

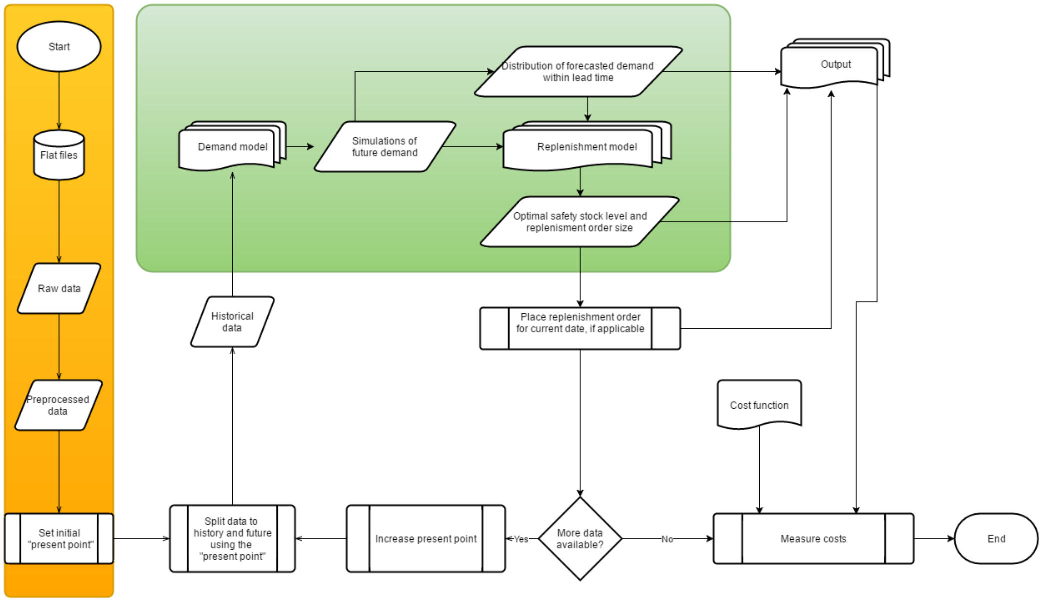
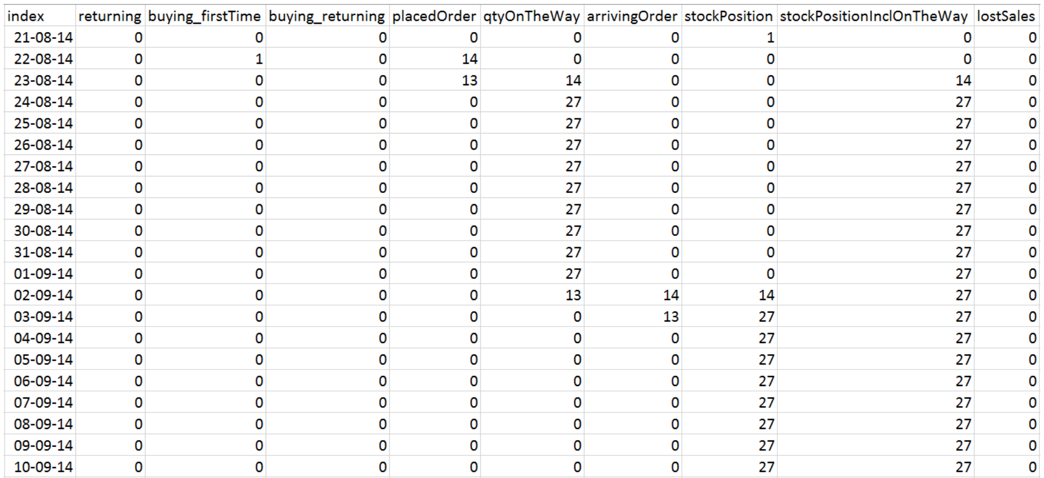

Start
Flat files
Raw data
Preprocessed data
Set initial "present point"
Demand model
Simulations of future demand
Distribution of forecasted demand within lead time
Replenishment model
Optimal safety stock level and replenisment order size
Output
Historical data
Place replenishment order for current date, if applicable
Cost function
Split data to history and future using the "present point"
Increase present point
Yes
More data available?
No
Measure costs
End

Figure 3.3: Flow diagram of the simulation process.



index	returning	buying_firstTime	buying_returning	placedOrder	qtyOnTheWay	arrivingOrder	stockPosition	stockPositionInclOnTheWay	lostSales
21-08-14	0	0	0	0	0	0	1	0	0
22-08-14	0	1	0	14	0	0	0	0	0
23-08-14	0	0	0	13	14	0	0	14	0
24-08-14	0	0	0	0	27	0	0	27	0
25-08-14	0	0	0	0	27	0	0	27	0
26-08-14	0	0	0	0	27	0	0	27	0
27-08-14	0	0	0	0	27	0	0	27	0
28-08-14	0	0	0	0	27	0	0	27	0
29-08-14	0	0	0	0	27	0	0	27	0
30-08-14	0	0	0	0	27	0	0	27	0
31-08-14	0	0	0	0	27	0	0	27	0
01-09-14	0	0	0	0	27	0	0	27	0
02-09-14	0	0	0	0	13	14	14	27	0
03-09-14	0	0	0	0	0	13	27	27	0
04-09-14	0	0	0	0	0	0	27	27	0
05-09-14	0	0	0	0	0	0	27	27	0
06-09-14	0	0	0	0	0	0	27	27	0
07-09-14	0	0	0	0	0	0	27	27	0
08-09-14	0	0	0	0	0	0	27	27	0
09-09-14	0	0	0	0	0	0	27	27	0
10-09-14	0	0	0	0	0	0	27	27	0

Figure 3.4: Example of a simulation run where the first three columns are the given data and the rest is gradually filled in during the simulation run.

ular day. We set two minimum limits to the metric. Firstly, we required this metric to be at least 14 days, as the demand distributions during the first days of the simulation are always very wide but, obviously, the stabilization has not yet taken place. Additionally, in order to limit the spread of the distribution, we required that the rolling coefficient of variation of 14 days of the median of the distribution should be below 0.8. We should emphasize here that the metric assumes that the demand has no trend and no seasonality. With our data, the assumption seems to be appropriate because only two of 356 items never stabilized.

We define the metric of stabilization to be the smallest t such that

- $D_{L,0.05,t} \leq D_{L,0.5,u} \leq D_{L,0.95,t}, u \geq t$ and
- $\frac{\sigma d}{\bar{d}} \leq 0.8, d \in \{D_{L,0.5,t-s} : s \in \{0, \ldots, 14\}\}$ and
- $t \geq 14$.

3.5 MCMC burn-in and sample size analysis

Necessity During the research, we noticed that, on average, Poisson distribution was far superior to negative binomial distribution. After some investigation, it appeared that the reason was the stability of the MCMC algorithm. During some runs it would result in reasonable forecasts, while during other runs wildly varying results were shown. For instance, if during the first ten days of sales history only a few items were sold, it is unreasonable to forecast that tens of thousands of items will be sold in the coming week. In MCMC algorithms, the two important parameters are the sample size and the burn-in percentage: the number of samples discarded as not being representative for the stable distribution. We needed to analyze the effect of the two variables on the result of the MCMC algorithm.

Analysis We decided to run a test to find appropriate values. We took a single policy and a single item, and ran simulations 10 times for each combination of sample size and burn-in values. For the burn-in, we chose values 0.1, 0.25 and 0.5. For the sample size analysis, we chose 500, 2,000 and 5,000.

Results In Table 3.10, we present the results of the analysis. We show the mean total costs per sample size and burn-in percentage. We saw that 5,000 should be an appropriate value for sample size and 10 percent for burn-in. Larger sample sizes would be preferred for accuracy of distribution estimation; however, that results in penalty in computation time. For that reason, we decided to go for a trade-off where we would use larger sample sizes at the beginning of the simulation and used smaller values later on. Specifically, from the first day in the simulation to the 14th day, we exponentially decreased sample size from 20,000 to 5,000 and keep it at 5,000 afterwords. A burn-in value of 10% was used throughout the simulation.

Samples\Burn-in %	0.1	0.25	0.5
500	4068.59	485.09	1549.49
2000	228.21	36484.68	195.42
5000	215.93	220.02	203.57

Table 3.10: Mean total cost per sample size and burn-in percentage.

Chapter 4

Results and recommendations

Chapter flow In this chapter, we will describe how we executed the simulation of using the particular replenishment policies and how we used the simulations to compare the policies. We will continue with some typical and interesting examples of simulation runs. Next, we will present the results and an analysis of the winning policy. We will conclude with a paragraph containing advice to Slimstock.

4.1 Simulation process and examples

Process The whole research process took place over the period from January 2016 until September 2016. Most of the time was consumed by analysis and preparation for the final calculation. For each item, replenishment policy and demand distribution type combination, we ran a simulation over the history of the item sales data. As already described in Section 3.3, for each of the days we created a model based on the historical data, simulated the future demand using the model and then used the simulated data in the replenishment model to make a decision about how large a replenishment order should be placed. At the end of the simulation, we were able to calculate cost statistics and the point where the demand distribution stabilized. Later, we aggregated costs across policies, and aggregated the stabilization metrics across demand distribution type. After months of preparation, it took ten nights of computing to calculate for each of the 456 items the simulations of 13 policies. The simulations resulted in 5,928 files containing daily information about forecasted demand, decisions of the policy and stock positions.

Examples Next, we will present some examples of the simulations. We start with the extreme, the highest cost case, which is shown in Figure 4.1. Here we deal with a 95% quantile policy with negative binomial demand distribution. The reason why this particular simulation has such a high cost is that, at some point during the simulation, the demand forecast apparently returns ridiculous values. This is most likely due to an issue already discussed in Section 3.5 - the stability of the MCMC algorithm. Next, lowest-cost simulation is shown in Figure 4.2. Here, we deal with a multi-order policy and also with negative binomial demand distribution. In this case, we have an initial stock that is large enough to last until the end of the simulation. No replenishment orders are placed during the simulation. This means that there are no lost sales costs, no order costs and minimal holding costs. Lastly, in Figure 4.3, we show a typical run that we encountered frequently. This particular simulation is an example of multi-order policy with negative binomial demand. In these sorts of simulations, an unrestricted policy initially overestimates the demand and thus suffers from high holding costs. As already discussed in earlier sections, this sort of behavior is caused by the fact that estimated demand distributions in the beginning of the simulations are very wide, due to a great deal of uncertainty.

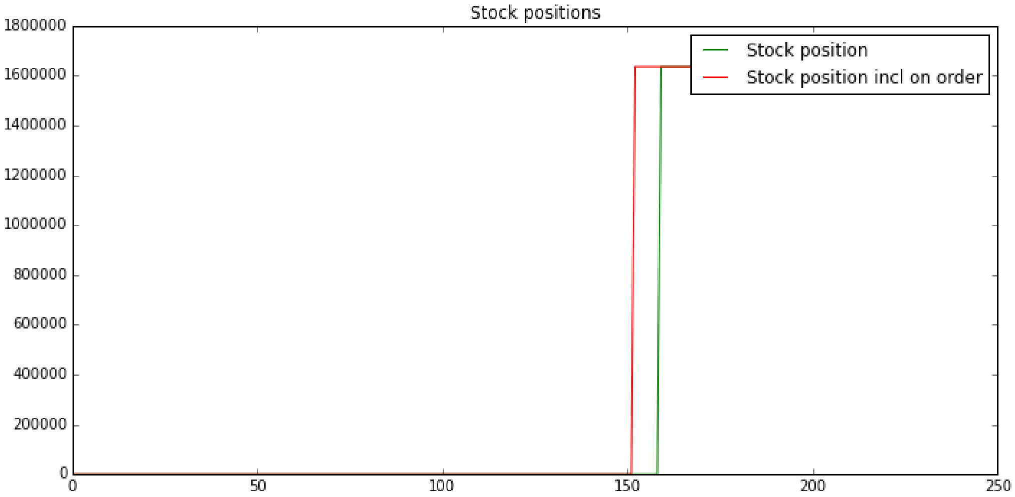


Figure 4.1: Worst-case simulation showing the stock position during the simulation run. One huge order is placed.

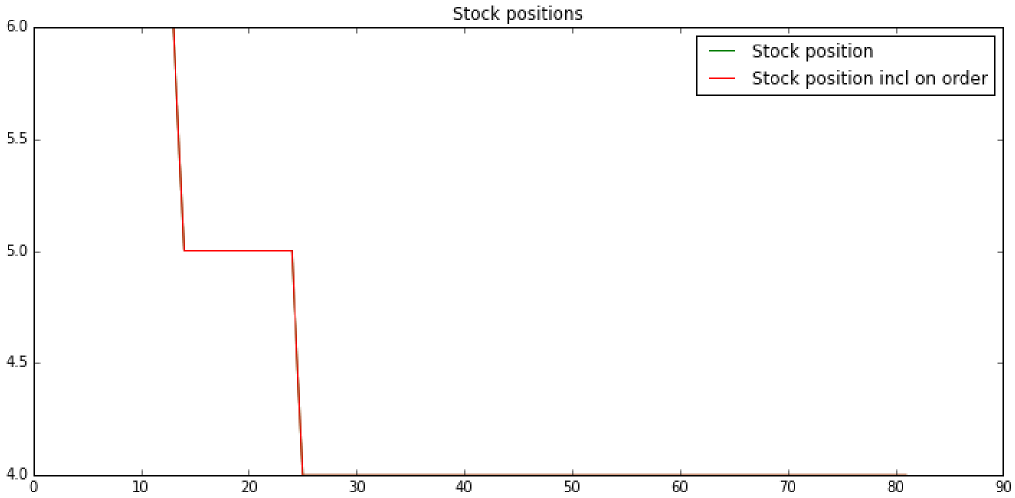


Figure 4.2: Best-case simulation showing the stock position during the simulation run. No orders are placed.

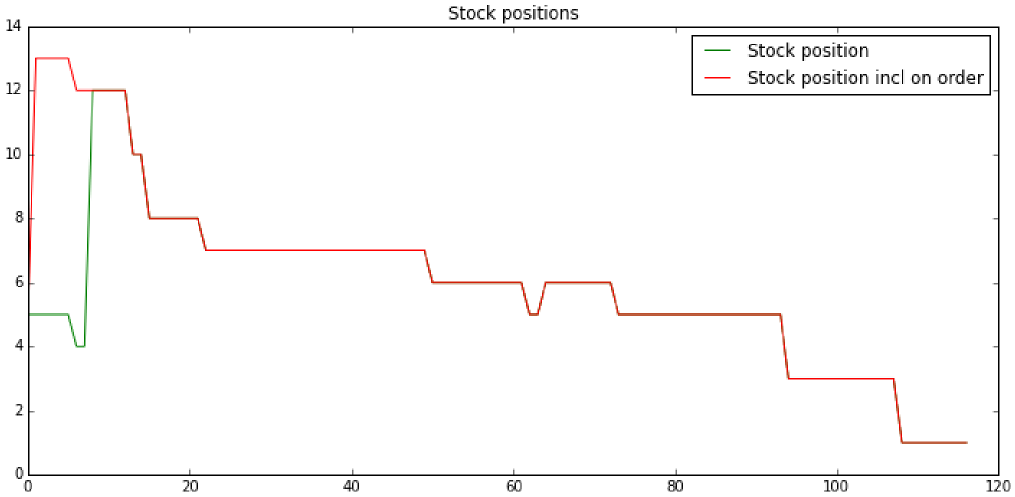


Figure 4.3: Example of simulation where unrestricted policy overestimates demand. Figure shows the stock position during the simulation run.

4.2 Results and analysis of winning policy

Results of simulation Table 4.1 shows final results of the simulations per policy and demand distribution. Figures 4.4 and 4.5 show the same information in graph form. We can see that the winning policy is the multi-order policy, independent of the demand distribution. A restricted policy using Poisson distribution is in an honorable third place. It is also notable that several of the policies using negative binomial distribution have higher costs than the baseline policy taking random decisions.

	Holding cost(std)	Holding cost(Mean)	Lost sales cost(std)	Lost sales cost(Mean)	Order cost(std)	Order cost(Mean)	Total cost(Mean)
Multi order_NegBin	486.07	136.83	792.06	171.99	0.61	0.27	309.09
Multi order_Poisson	474.45	133.18	1017.95	184.29	1.02	0.43	317.90
OrderUpTo dynamic quantile restricted_Poisson	502.01	113.20	865.39	227.43	1.06	0.26	340.89
OrderUpTo dynamic quantile Q=halfS_Poisson	739.00	185.82	804.69	164.19	0.33	0.16	350.17
OrderUpTo dynamic quantile_Poisson	1129.22	288.63	630.41	132.36	0.44	0.17	421.16
OrderUpTo dynamic quantile restricted_NegBin	2011.54	207.01	926.39	227.74	0.95	0.23	434.98
OrderUpTo dynamic quantile Q=S_Poisson	1122.63	286.95	936.08	154.05	0.20	0.12	441.12
OrderUpTo fixed quantile_Poisson	1305.03	299.21	641.52	150.54	0.24	0.11	449.86
Random_None	1378.27	556.64	1662.74	153.99	0.64	0.63	711.26
OrderUpTo dynamic quantile Q=halfS_NegBin	7851.61	587.59	811.11	161.37	0.30	0.15	749.11
OrderUpTo dynamic quantile_NegBin	15677.93	1093.71	700.00	138.99	0.36	0.15	1232.85
OrderUpTo dynamic quantile Q=S_NegBin	15678.89	1094.67	948.43	153.73	0.20	0.13	1248.53
OrderUpTo fixed quantile_NegBin	21637.01	1169.43	1025.89	225.25	0.12	0.04	1394.72

Table 4.1: Costs per policy.

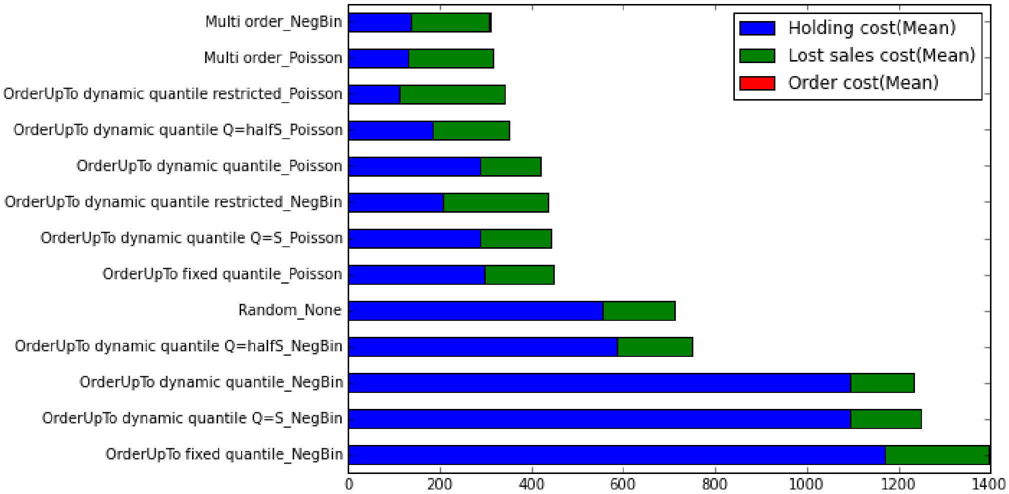


Figure 4.4: Mean cost per policy

Demand distribution The fact that the winning policy took two of the top positions was slightly surprising. We could see that in general, per pol-

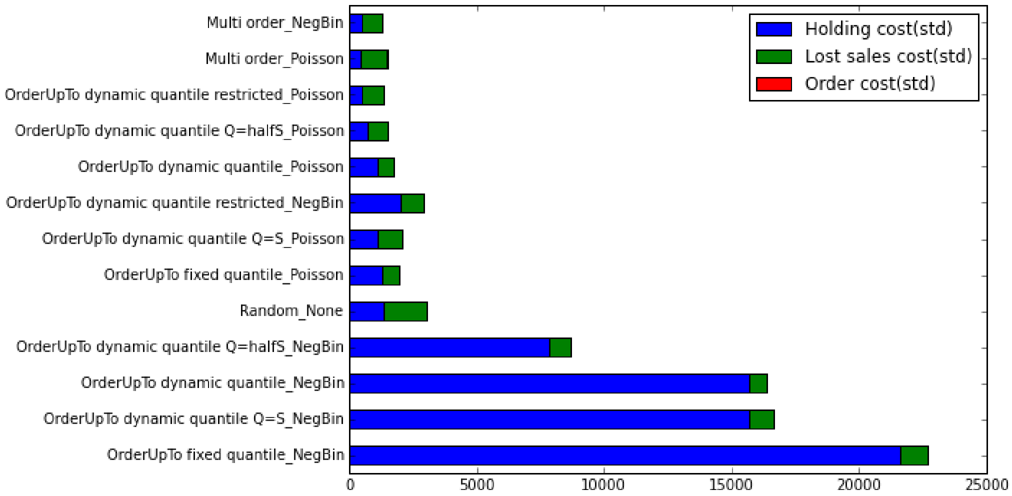


Figure 4.5: Standard deviation of cost components per policy

icy, Poisson distribution outperformed negative binomial distribution. We suspected that this is related to the instability of the MCMC algorithm and not because that Poisson distribution would fit the data better. In general, although the winner of our test was negative binomial distribution, Poisson distribution should be preferred because using the multi-order policy, it delivers equivalent results and is computationally far cheaper.

Analysis of multi-order policy So, how could negative binomial distribution outperform Poisson distribution with multi-order policy if, in general, Poisson distribution is superior? We ran a test on 153 random items and checked what quantiles were used during the simulations. Multi-order policy used on average 0.12 quantile with 0.07 standard deviation, whereas dynamic quantile $\frac{c_{ls}}{c_{ls}+c_h}$ from (3.5) would have used 0.51 quantile with 0.18 standard deviation. Multi-order policy uses a significantly lower quantile than the other policies. This also makes sense as multi-order policy calculates s, reorder level and other policies calculate S, order-up-to level. Using lower quantile makes multi-order policy less sensitive to the extreme forecasts that the MCMC algorithm occasionally produces.

Stabilization Table 4.2 shows the results of the analysis of stabilization. With negative binomial distribution, stabilization occurs approximately on the 21st day and with Poisson distribution on the 28th day. Only two of all the simulations never stabilized. According to Slimstock, the demand distribution stabilization period was the most useful result of the research.

	Mean	std
NegBin	21.37	20.56
Poisson	27.53	36.90

Table 4.2: First day of stabilization of demand distribution.

4.3 Advice

Advice The sponsor of this research project was Slimtock B.V., which provided the database as well as guidance, and posed the research question. Our advice to Slimstock would be to consider implementing multi-order policy in their software, as the implementation should be relatively straightforward and its usage could be beneficial during the early phases of a product life cycle. Multi-order policy could be used until the end of the period when demand is estimated to be stable. For computational reasons, Poisson distribution should be used for modeling of the demand. Also, the framework that has been built for the research could be used to compare any policies that Slimstock might consider using.

Chapter 5

Conclusions and future work

Chapter flow In this final chapter, we will first present the potential relevance of this research for the scientific and business community. Next, we will discuss several issues with the current research, as well as possible solutions. Additionally, we will describe several topics that could be included in future research on the same topic.

5.1 Relevance and problems

Relevance The results are applicable to the business community as the methodology used for testing the policies is clearly described and could be easily implemented. Numerical comparison of replenishment policies can also be interesting from a scientific point of view, as it provides an empirical study of the policies based on theoretical models. The sponsor of the research, Slimstock, found the results useful and has adjusted the development roadmap to include the winning policy in their software.

Order costs As could be seen from the results of the simulations, order costs were negligible compared to holding and lost sales costs. This fact makes us question whether the fixed value of order cost used in the current research was correct. In future research, this constant should be considered carefully.

MCMC One of the greatest problems we faced during the research was the stability of the MCMC algorithm that was used to estimate the posterior distribution of the parameters of negative binomial demand. We tried to overcome the stability issues by increasing the number of samples of the MCMC algorithm, but still, negative binomial distribution was inferior to Poisson distribution in most cases. We are unsure whether this sort of instability was caused by a bug in the package that was used in computations or because it is inherent to the methodology that we used.

5.2 Future work

Unobserved lost sales One of the criticisms to the current research would be not taking into account the unobserved lost sales when estimating demand. In this study, this was purely the result of the given data set which did not include any information about the true stock position, thus, making it impossible to know when the shop ran out of stock. In the future, appropriate data should be obtained in order to allow such analysis.

Loss items During the filtering of the initial data set, the items whose sales price was less than their cost price were removed. The reason was that the formulas that were used did not support such items. In accordance with the policies, such items should not be sold. In reality, of course, such items are being sold. Motivation behind such intentional loss making can vary. For instance, such items can be sold for promotional reasons. Additionally, an item could have to be included with another profit-making item. Replenishment models presented in the current paper do not cover such items and future research could investigate the topic.

Improvements When checking the simulations, we ran into an example in which even the winning strategy could be improved. As seen from Figure 5.1, at some point multi-order policy using Poisson demand, makes an obviously unreasonable order for an amount that will never be consumed. In future research, it would be interesting to see if a modified multi-order policy would outperform vanilla multi-order policy. One possible way would be to restrict it by some value as described in (2.7). Another way would be to introduce an ordering decision as described in (2.10).

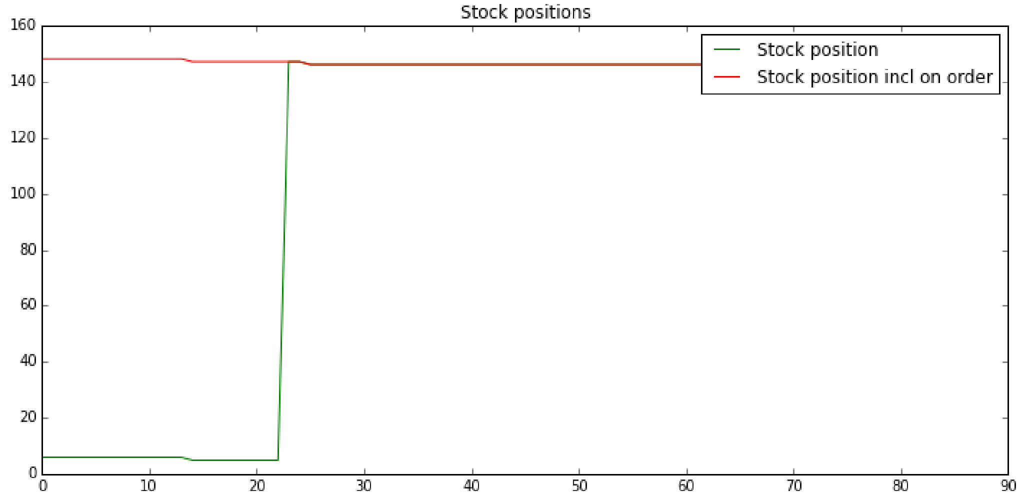


Figure 5.1: Multi-order policy fails by placing a huge order. Figure shows stock position during simulation run.

Bibliography

[1] Bayesian conjugate prior. https://en.wikipedia.org/wiki/Conjugate_prior. [Online; accessed 05-July-2016].

[2] Eclipse. https://www.eclipse.org. [Online; accessed 05-July-2016].

[3] Metropolis-Hastings algorithm. https://pymc-devs.github.io/pymc/theory.html?highlight=metropolis#the-metropolis-hastings-algorithm. [Online; accessed 05-July-2016].

[4] Pandas. http://pandas.pydata.org/. [Online; accessed 05-July-2016].

[5] Pydev. http://www.pydev.org/. [Online; accessed 05-July-2016].

[6] Pymc. https://pymc-devs.github.io/pymc/. [Online; accessed 05-July-2016].

[7] M. Bijvank. *Service Inventory Management.* 2009.

[8] L. Chen. Bounds and heuristics for optimal Bayesian inventory control with unobserved lost sales. *Operations Research*, 2010.

[9] W. Powell D. Negoescu, P. Frazier. Optimal learning policies for the newsvendor problem with censored demand and unobservable lost sales. Technical report.

[10] R.E. Walpole et al. *Essentials of probability and statistics for engineers and scientists.* 2013.

[11] Python Software Foundation. Python. https://www.python.org/. [Online; accessed 05-July-2016].

[12] G. van Houtum J. Arts, R. Levi and B. Zwart. Base-stock policies for lost-sales models: Aggregation and asymptotics. Technical report, 2015.

[13] G. Koole. *Optimization of Business Processes*. 2013.

[14] J.G. Shanthikumar L.H. Liyanage. A practical inventory control policy using operational statistics. *Operations Research Letters*, 2005.

[15] S.G. Johansen M. Bijvank. Periodic review lost-sales inventory models with compound poisson demand and constant lead times of any length. *European Journal of Operational Research*, 2012.

[16] E.L. Porteus M.A. Lariviere. Stalking information: Bayesian inventory management with unobserved lost sales. *Management Science*, 2015.

[17] T.E. Morton. The near myopic nature of the lagged proportional cost inventory problem with lost sales. *Operations Research*, 1971.

[18] S.A. Smith N.Agrawal. Estimating negative binomial demand for retail inventory management with unobservable lost sales. *Naval Research Logistics*, 1996.

[19] P.H.Zipkin Eds. S.C. Graves, A.H.G. Rinnooy Kan. *Logistics of production and inventory*. 1993.

[20] A. Thorstenson S.G. Johansen. Pure and restricted base-stock policies for the lost-sales inventory system with periodic review and constant lead times. Technical report, 2008.

[21] K. Zhu X. Lu, J. Song. Inventory control with unobservable lost sales and Bayesian updates. *Preprint*, 2006.