coordinated\_ordering12

GroupB

06/08/2020

Table of Contents

# Data preparation

, , , ,

Storage Information

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Total\_racks | levels\_per\_rack | rack\_length | rack\_width | rack\_height |
| Rack Information | 8 | 4 | 6000 | 1750 | 300 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N0 | material ID | demand per day | box ID | pieces/box | price |
| 1 | 7305667+74 | 12 | 6203060 | 45 | 1.51 |
| 2 | 7305669+77 | 30 | 6203059 | 15 | 7.62 |
| 3 | 7305670+77 | 30 | 6203059 | 15 | 1.62 |
| 4 | 7305673+76 | 30 | 6203059 | 16 | 1.51 |
| 5 | 7305674+76 | 30 | 6203059 | 16 | 1.51 |
| 6 | 7305817+74 | 60 | 6203060 | 16 | 4.55 |
| 7 | 7305819+79 | 30 | 6203059 | 6 | 11.13 |
| 8 | 7305820+79 | 30 | 6203059 | 6 | 2.86 |
| 9 | 7305823+73 | 30 | 6203059 | 30 | 4.05 |
| 10 | 7305824+73 | 30 | 6203059 | 30 | 2.56 |
| 11 | 7305832+73 | 30 | 6203061 | 30 | 1.50 |
| 12 | 7305833+73 | 30 | 6203059 | 12 | 2.02 |
| 13 | 7305834+73 | 30 | 6203059 | 12 | 4.27 |
| 14 | 7305863+75 | 60 | 6203059 | 16 | 1.52 |
| 15 | 7305865+77 | 30 | 3106410 | 14 | 3.15 |
| 16 | 7305867+76 | 60 | 6203059 | 48 | 1.58 |
| 17 | 7305971+75 | 6 | 6203059 | 15 | 3.32 |
| 18 | 7305972+75 | 6 | 6203059 | 15 | 3.13 |
| 19 | 7305977+74 | 24 | 6203061 | 36 | 1.54 |
| 20 | 7306005+73 | 6 | 3108210 | 20 | 1.50 |
| 21 | 7306006+73 | 6 | 3108210 | 20 | 3.15 |
| 22 | 7306009+75 | 6 | 6203060 | 13 | 3.62 |
| 23 | 7306010+75 | 6 | 6203060 | 13 | 2.13 |
| 24 | 7306013+73 | 6 | 6203060 | 16 | 1.53 |
| 25 | 7306014+73 | 6 | 6203060 | 16 | 1.70 |
| 26 | 7306017+79 | 6 | 6203059 | 9 | 1.51 |
| 27 | 7306018+79 | 6 | 6203059 | 9 | 6.62 |
| 28 | 7306023+78 | 30 | 6203060 | 15 | 2.13 |
| 29 | 7306024+78 | 30 | 6203060 | 15 | 2.15 |
| 30 | 7306025+75 | 6 | 6203061 | 20 | 1.85 |
| 31 | 7306026+75 | 6 | 6203061 | 20 | 1.62 |
| 32 | 7306033+77 | 6 | 6203061 | 30 | 2.53 |
| 33 | 7306034+77 | 6 | 6203061 | 30 | 2.11 |
| 34 | 7306037+79 | 6 | 6203060 | 20 | 3.82 |
| 35 | 7306038+30 | 6 | 6203060 | 20 | 2.28 |
| 36 | 7308013+79 | 30 | 3108210 | 13 | 19.46 |
| 37 | 7308014+79 | 30 | 3108210 | 13 | 1.58 |
| 38 | 7313474+73 | 30 | 6203061 | 8 | 2.82 |
| 39 | 7313981+31 | 6 | 6203059 | 12 | 5.68 |
| 40 | 7313982+31 | 6 | 6203059 | 12 | 9.67 |
| 41 | 7326393+74 | 30 | 6203059 | 12 | 1.75 |
| 42 | 7326394+74 | 30 | 6203059 | 12 | 2.96 |
| 43 | 7326409+74 | 30 | 6203060 | 11 | 1.87 |
| 44 | 7326825+72 | 30 | 6203060 | 18 | 1.68 |
| 45 | 7326826+72 | 30 | 6203060 | 18 | 1.69 |
| 46 | 7328209+73 | 30 | 6203061 | 16 | 1.87 |
| 47 | 7328210+73 | 30 | 6203061 | 16 | 1.76 |
| 48 | 7332421+76 | 30 | 6203062 | 5 | 3.45 |
| 49 | 7332422+76 | 30 | 6203062 | 5 | 1.57 |
| 50 | 7343137+72 | 30 | 6203059 | 32 | 6.74 |
| 51 | 7343138+74 | 30 | 6203059 | 32 | 1.96 |
| 52 | 7346592+72 | 360 | 3103147 | 64 | 1.99 |
| 53 | 7346813+73 | 60 | 6203061 | 30 | 2.45 |
| 54 | 7348808+73 | 30 | 3103147 | 18 | 2.58 |
| 55 | 7355005+72 | 30 | 6203061 | 36 | 1.67 |
| 56 | 7355006+72 | 30 | 6203061 | 36 | 3.08 |
| 57 | 7455633+73 | 24 | 3108210 | 8 | 24.57 |
| 58 | 7455634+73 | 24 | 3108210 | 8 | 1.60 |
| 59 | 7455687+72 | 24 | 3108210 | 24 | 3.99 |
| 60 | 7455688+72 | 24 | 3108210 | 24 | 10.70 |
| 61 | 7455691+71 | 24 | 3104147 | 20 | 4.39 |
| 62 | 7455692+71 | 24 | 3104147 | 20 | 4.39 |

## Convert to Boxes

Convert demand , quantity and unit price all from part to box which is the capacity of a box for part .

Demand for box with part .

Price for box with material .

value of box

Merged data with yearly demand

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| box ID | N0 | material ID | box\_cost | ordering\_cost |
| 3103147 | 54 | 7348808+73 | 46.4 | 50 |
| 3103147 | 52 | 7346592+72 | 127.4 | 50 |
| 3104147 | 61 | 7455691+71 | 87.8 | 45 |
| 3104147 | 62 | 7455692+71 | 87.8 | 45 |
| 3106410 | 15 | 7305865+77 | 44.1 | 60 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| box ID | N0 | material ID | demand per day | box\_cost | ordering\_cost |
| 6203060 | 1 | 7305667+74 | 12 | 67.95 | 75 |
| 6203059 | 2 | 7305669+77 | 30 | 114.30 | 80 |
| 6203059 | 3 | 7305670+77 | 30 | 24.30 | 80 |
| 6203059 | 4 | 7305673+76 | 30 | 24.16 | 80 |
| 6203059 | 5 | 7305674+76 | 30 | 24.16 | 80 |

# Separate Ordering (SO)

ordering cost for part

stock holding cost rate based on unit price and interest rate therefore

## EOQ modelling

calculate EOQ for each part i:

$$ q\_i^{max} = \sqrt \frac{2 \cdot y\_i \cdot (c\_i^{or}+c^{-or})}{pr\_i \cdot h}$$

Min EOQ, used in starting feasible solution.

$$ q\_i^{min} = \sqrt \frac{2 \cdot y\_i \cdot (c\_i^{or})} {pr\_i \cdot h}$$

these notations can be used interchangably both meaning Maximum EOQ.

Both EOQs added to the table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| box ID | N0 | material ID | demand per day | ordering\_cost | eoq.min | eoq.max |
| 6203060 | 1 | 7305667+74 | 12 | 75 | 39 | 180 |
| 6203059 | 2 | 7305669+77 | 30 | 80 | 86 | 381 |
| 6203059 | 3 | 7305670+77 | 30 | 80 | 186 | 825 |
| 6203059 | 4 | 7305673+76 | 30 | 80 | 181 | 802 |
| 6203059 | 5 | 7305674+76 | 30 | 80 | 181 | 802 |

## Lane Occupation

sorting column lenght or width.This determines the lane width

if sorted by length then the value is width and vice versa, used for rack lenght.

Add and to product\_data table

Both EOQs added to the table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| N0 | material ID | ordering\_cost | eoq.min | eoq.max | b\_sorting | b\_not\_sorting |
| 1 | 7305667+74 | 75 | 39 | 180 | 396 | 297 |
| 2 | 7305669+77 | 80 | 86 | 381 | 594 | 396 |
| 3 | 7305670+77 | 80 | 186 | 825 | 594 | 396 |
| 4 | 7305673+76 | 80 | 181 | 802 | 594 | 396 |
| 5 | 7305674+76 | 80 | 181 | 802 | 594 | 396 |

E.g Using box\_ID 6203060, sorted by length , Also for material ID 7305667+74 with therefore,

*1.* Number of boxes in a lane:

*2.* How many lanes part will occupy if you order some number of boxes

$$lane\_i=q\_i \cdot \frac {b\_i^{-1}}{rack\_{length}}=180 \cdot \frac {297}{6000}=8.91= 9 \space lanes $$

number of lanes per item

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| lanes | 9 | 26 | 55 | 53 | 53 | 33 | 52 | 103 |

1. Do we meet the capacity contraint?

*3.* Total number of lanes constraints

Collapsing Rack 4 levels in a rack and joining the 8 racks to become 1 level.

Total rack width available:

Note: rack\_total\_width which is 56,000 mm, looking at the 9 patterns all of which have 150 mm in waste, meaning all 32 levels will have 150 mm waste each. Therefor thereby changing rack total width 51,200.

These *9 patterns* will be explained later in the chapter.

Number of lanes that minimizes cost

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Min lane | 2 | 6 | 13 | 12 | 12 | 8 | 12 | 24 | 4 | 5 | 4 | 14 |
| Max lane | 9 | 26 | 55 | 53 | 53 | 33 | 52 | 103 | 18 | 22 | 22 | 61 |

Min lane: This lanes was derived from .

Max lane: derived from .

* *Overflow*

This is how much we are lacking in capacity assuming we considered optimum lanes thus optimum quatities for all our items when placing an order.

* Overflow in mm

Expressing it in mm terms

- relative overflow in %

In percentage terms

Rack Capacity overflow in mm and percentage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | mm\_lane\_min | mm\_lane\_max | % lane\_min | % lane.max |
| Overflow | 218694 | 1173092 | 427.1367 | 2291.195 |

*As seen above*, the capacity is exceeded by lots of margin even without adding overall ordering capacity it still exceeds by

* *Introducing capacity constraint*

x

269894 mm is the total expected to be occupied

|  |
| --- |
| x |
| Capacity constraint violated by: 218694 |

As seen above contraint was voilated by , thus we don’t have that kind of capacity for that, therefore we need to optimize to fit our capacity.

Using our example to get number of the total width i.e (summation of lanes) in : Summation of lanes can simply be:

Using equation (1) to prove this concept.

1. Adjust Q by reducing it:

number of orders for part .

average inventory for part

## Constrained EOQ optimization

Subject to:

## ROI: R Optimization Infrastructure

## Registered solver plugins: nlminb, alabama, deoptimr, deoptim, glpk, quadprog, symphony.

## Default solver: auto.

Cost Values and capacities

|  |  |  |
| --- | --- | --- |
|  | Cost Function (€) | Capacity in (mm) |
| Eoq Max | 172279.7 | 1224292 |
| Eoq Min | 124501.5 | 269894 |
| As seen ab | ove, capcities of bo | th EOQ’s exceeds the our rack capacity of 56,000 mm. |

Now adding capacity constraint

## [1] TRUE

Capacitated values for cost and Quantity

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cost | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 205079.8 | 16 | 20 | 20 | 20 | 20 | 19 | 15 | 15 | 18 | 19 | 18 | 20 | 20 | 19 |

removing waste of 150 for every 32 levels

## No constraint

## Compare results

costs

|  |  |  |
| --- | --- | --- |
|  | Cost Function (€) | capacitaty |
| Eoq Max | 172279.7 | 1224292 |
| Optimized Q | 205079.8 | 50340 |

Optimum costs with no capacity constraint using EOQ.max considers is €172279.7, with no capacity constraint optimized using EOQ.min is €124508.6, with capacity constraint, it costs €205079.8 This cost is expected to be higher when capacity constraint is included, as limited capacity of the rack will not allow to take advantage of cost savings.

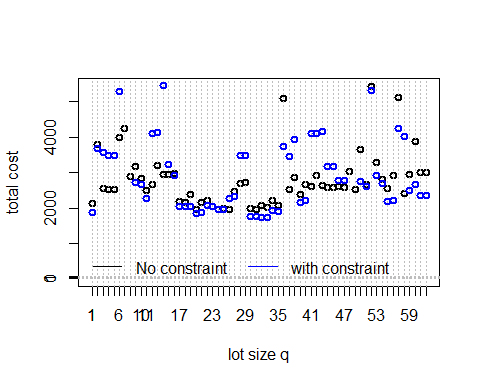
* Plot for lot sizes of costs EOQ Max vs Optimized for capacity

cost function per item

|  |
| --- |
| x |
| It costs € 32800 extra due to lack of capacity |

The above result shows that If there is no capacity constraint, the cost will be lower.

|  |
| --- |
| x |
| 26 Items costs more or same when capacity becomes a problem |



* Notice that 26 blue dots which represents the more expensive items when capacity is accounted for.
* Also the notice , for most items where the blue is larger than black (non constraint items)(the variance is much more) the difference is much larger. the reverse is the case when black is larger than blue.

# Joint Ordering (JO)

Using Joint replenishment problem

Assumptions:

* One supplier with outbound storage.
* products
* Demand rates:
* Stock\_holding cost rates:
* Specific setup costs:
* General setup costs:
* Cycle time of product :

## Objective Function with no capacity constraint

* basic cycle time
* Holding cost multiplier are $H\_0 \space and \space H\_i$
* subject to:

–> an important side note is that the JRP is basically a quite standard EOQ problem which uses the substitution , i.e., . Then, you substitute ans result in the JRP (adding the common ordering cost).

* *Cycle Time*

we determine and

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 52 | 57 | 36 | 7 |
|  | 50.0000000 | 65.0000000 | 65.0000000 | 80.0000000 |
|  | 9386.4320000 | 7724.8080000 | 7652.6450000 | 4374.0900000 |
|  | 0.0729852 | 0.0917303 | 0.0921618 | 0.1352387 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 52 | 57 | 36 | 7 |
|  | 0.0053268 | 0.0084144 | 0.0084938 | 0.0182895 |
|  | 0.1651320 | 0.0943824 | 0.0678407 | 0.0604023 |

$$B=\sqrt \frac {\sum\_{i=0}^n \frac {c\_i^{or}}{m\_i}}{\sum\_{i=0}^n m\_i \cdot H\_i} $$

## [1] 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38  
## [26] 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 5 | 2 | 57 | 36 | 7 | 60 | 6 |
|  | 0.8 | 1.01 | 1.01 | 1.49 | 1.53 | 1.59 |
|  | 1.0 | 1.00 | 1.00 | 1.00 | 2.00 | 2.00 |
|  | 1.0 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

break occured after 13

Reoptimizing basic cycle time yields and the total cost are 4.99433410^{4}

The order quantities are given by multiplying the cycle times with demand rates , i.e.  such that

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 52 | 57 | 36 | 7 | 60 | 6 |
|  | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
|  | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 |
|  | 133.87 | 71.38 | 54.95 | 118.97 | 23.79 | 89.28 |

–> Now the question is: Is this feasible? –> no:

cost with no capacity constraint

|  |  |  |
| --- | --- | --- |
|  | Cost(€) | Capacity (mm) |
| Not constrained | 49943 | 281834 |

Though there is alot of cost savings, but the capacity required to meet this cost far exceeds the current capacity on ground.

When you try to optimize the JRP directly, without introducing the substitution you need to update the JRP function as follows

cost with no capacity constraint using cycle Time

|  |  |  |
| --- | --- | --- |
|  | Cost(€) | Capacity (mm) |
| Not constrained | 151952 | 1104014 |

Compared to the previous solution, this is far more inefficient.As It costs more and also required more capacity.

## Including capacity constraint

–> You should integrate the shelf capacity constraint while still using basic period approach as it so far has the lowest cost when not constrained.

This is formally defined as

Thus, with the substitution above, the left-hand side changes to

Thus, you can change the constraint function quite easily

Then, you can use the original JRP function and update the model by including the capacity constraint. Therefore I recommend that you fix to some appropriate value based on the feasible solution above

cost with capacity constraint using Basic period approach

|  |  |  |
| --- | --- | --- |
|  | Cost(€) | Capacity (mm) |
| Capacitated | 268254 | 51132 |
| Though more ex | pensive, t | his met the capacity. |

First 7 items in order before reordering

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 52 | 57 | 36 | 7 | 60 | 6 |
|  | 1.00 | 2.00 | 3.00 | 1.00 | 7.00 | 2.00 |
|  | 0.01 | 0.02 | 0.03 | 0.01 | 0.07 | 0.02 |
|  | 14.00 | 15.00 | 18.00 | 13.00 | 18.00 | 19.00 |

First 7 items in order After reordering

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
|  | 24.00 | 4.00 | 4.00 | 4.00 | 4.00 | 2.00 |
|  | 0.23 | 0.04 | 0.04 | 0.04 | 0.04 | 0.02 |
|  | 16.00 | 20.00 | 20.00 | 19.00 | 19.00 | 19.00 |

* *Order by m*

After reordering

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|  | 16 | 20 | 20 | 19 | 19 | 19 | 13 | 13 | 18 |

find a layout scheme such that a precise shelf layout results.

calculate the number of lanes required for each product given a certain order quantity (integer number of boxes):

whereby is the number of boxes per lane dedicated to product . I.e.,

. Thus, for the solution of the JRP we have:

number of lanes for each items

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| lanes | 1 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 2 |

Now, these lanes have to be assigned to the levels of the shelves. Therefore, we first need to determine the types of lanes required. The lanes are just described by their width. Due to the safety margins we should round the lane width to full centimeter. I.e., we need to assign lanes with the following widths, and

Now comes the tricky part: We have to decide how many lanes of a certain width should be assigned for each level. Luckily, there is only a small number of useful patterns of lanes per level. To be precise there are 9 efficient patterns to arrange these 3 lane types (assuming we use each level exhaustively):

Total of 9 patterns

|  |  |  |  |
| --- | --- | --- | --- |
|  | 200 | 400 | 600 |
| 1 | 8 | 0 | 0 |
| 2 | 0 | 4 | 0 |
| 3 | 0 | 1 | 2 |
| 4 | 2 | 0 | 2 |
| 5 | 6 | 1 | 0 |
| 6 | 4 | 2 | 0 |
| 7 | 2 | 3 | 0 |
| 8 | 3 | 1 | 1 |
| 9 | 1 | 2 | 1 |

Let indicate the number of lanes of type associated to pattern . Now we need to assess the number of lanes per required of each type. As outlined above, we know the number of lanes per product , thus we can deduce the demand for lane type () by summing up the for each lane type, i.e., :

number of lanes per required of each type

|  |  |  |  |
| --- | --- | --- | --- |
|  | 200 | 400 | 600 |
| 1 | 14 | 15 | 70 |
| This | misfit | was d | ue to approximations in the on lane assignment by making fraction value to ceiling. |

Need to reduce 70 units to 64 for type 600.

reduce lane assignment while fulfilling demand, to do this, we need to change some values to their floor values. but this can cause backorder, that is inability to meet demand.

lanes in fraction

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
|  | 0.8 | 1.333333 | 1.333333 | 1.266667 | 1.266667 | 0.95 |

To reduce the effect of backorder, as seen above, items 15, 17, 18, 50,57 and 58 are closest decimals to there floor.

selected items

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 15 | 17 | 18 | 50 | 57 | 58 |
|  | 1.066667 | 1.133333 | 1.133333 | 1.133333 | 2.142857 | 2.142857 |

, , , ,,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N0 m | aterial ID | b\_sorting | b\_not\_sorting | number\_of\_lanes |
| 1 | 7305667+74 | 396 | 297 | 1 |
| 2 | 7305669+77 | 594 | 396 | 2 |
| 3 | 7305670+77 | 594 | 396 | 2 |
| 4 | 7305673+76 | 594 | 396 | 2 |
| 5 | 7305674+76 | 594 | 396 | 2 |

# assign lane width as names  
names(l.start) <- round(product\_data$b\_sorting/100)\*100   
  
# number of items with lane types 200, 400, 600 that is demand for each lane type  
ld <- c(sum(l.start[names(l.start) == "200"]),  
sum(l.start[names(l.start) == "400"]),  
sum(l.start[names(l.start) == "600"]))  
kable(as.data.frame(ld),format = "pandoc",caption = "Number of lanes for 200, 400 and 600")

Number of lanes for 200, 400 and 600

|  |
| --- |
| ld |
| 14 |
| 15 |
| 64 |

* *Changes made for lane assignment*

1. rack\_total\_width which was 56,000 mm, looking at the 9 patterns all of which have 150 mm in waste, meaning all 32 levels will have 150 mm waste each. Therefor thereby changing rack total width 51,200.
2. These changes then affects the values of q for all the 62 items involved.

pat<-data.frame(patterns[3:4,])   
colnames(pat)<-c("200","400","600")  
  
kable(pat,caption = "Patterns needed", row.names = T, escape = F,format = "pandoc")

Patterns needed

|  |  |  |  |
| --- | --- | --- | --- |
|  | 200 | 400 | 600 |
| 1 | 0 | 1 | 2 |
| 2 | 2 | 0 | 2 |

* *Assign Patterns*

Pattern Assigned for each level

|  |  |  |  |
| --- | --- | --- | --- |
|  | 200 | 400 | 600 |
| 1 | 0 | 1 | 2 |
| 2 | 0 | 1 | 2 |
| 3 | 0 | 1 | 2 |
| 4 | 0 | 1 | 2 |
| 5 | 0 | 1 | 2 |
| 6 | 0 | 1 | 2 |
| 7 | 0 | 1 | 2 |
| 8 | 0 | 1 | 2 |
| 9 | 0 | 1 | 2 |
| 10 | 0 | 1 | 2 |
| 11 | 0 | 1 | 2 |
| 12 | 0 | 1 | 2 |
| 13 | 0 | 1 | 2 |
| 14 | 0 | 1 | 2 |
| 15 | 0 | 1 | 2 |
| 16 | 0 | 1 | 2 |
| 17 | 2 | 0 | 2 |
| 18 | 2 | 0 | 2 |
| 19 | 2 | 0 | 2 |
| 20 | 2 | 0 | 2 |
| 21 | 2 | 0 | 2 |
| 22 | 2 | 0 | 2 |
| 23 | 2 | 0 | 2 |
| 24 | 2 | 0 | 2 |
| 25 | 2 | 0 | 2 |
| 26 | 2 | 0 | 2 |
| 27 | 2 | 0 | 2 |
| 28 | 2 | 0 | 2 |
| 29 | 2 | 0 | 2 |
| 30 | 2 | 0 | 2 |
| 31 | 2 | 0 | 2 |
| 32 | 2 | 0 | 2 |
| sum | 32 | 16 | 64 |

* *Assign Items to racks*

## Illustration

|  |  |  |  |
| --- | --- | --- | --- |
| Mi= | 1 Mi= | 2 Mi= | 3 |
| 1 | 7,8,48,49,52 | 6,14,38,43,57,58 | 12,13,15,36,37,41,42,53 |
| 2 | 7,8,48,49,52 | 0 | 0 |
| 3 | 7,8,48,49,52 | 6,14,38,43,57,58 | 0 |
| 4 | 7,8,48,49,52 | 0 | 12,13,15,36,37,41,42,53 |
| 5 | 7,8,48,49,52 | 6,14,38,43,57,58 | 0 |
| 6 | 7,8,48,49,52 | 0 | 0 |

display Item 4 , having lotsize of 19

item 4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | eoq.min | eoq.max | b\_sorting | b\_not\_sorting | number\_of\_lanes | Order\_frequency\_M | Lot\_size\_q |
| 4 | 181 | 802 | 594 | 396 | 2 | 4 | 19 |
| \* \*D | emand vs Q | uantity\* |  |  |  |  |  |

which is the capacity of a box for part . Demand in Years for item Demand per day in boxes for item . Demand per day in items for item $yb\_i= $

cycle time is number of days it takes before a new order.

E.g Item 4 has lot size quantity of 19 boxes, and daily demand in boxes is 1.8

item 4 has order frequency M of 4, and is value is 9.9 then floor value is 9

* *Explain formular for demand per day in boxes*

item 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | number\_of\_lanes | Order\_frequency\_M | Lot\_size\_q | deman\_per\_day\_boxes | cycle\_time\_in\_days |
| 4 | 2 | 4 | 19 | 1.875 | 9.903517 |

As seen above, cycle time in days in 9.9 days, but we need to round this to the nearest floor

cycle time for each item

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |
| M | 24.00000 | 4.000000 | 4.000000 | 4.000000 |
| days | 59.44375 | 9.888421 | 9.888421 | 9.903517 |

give them floor values

cycle time for each item

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| M | 24 | 4 | 4 | 4 | 4 | 2 | 1 | 1 | 7 | 7 |
| days | 59 | 9 | 9 | 9 | 9 | 4 | 2 | 2 | 17 | 17 |

* *Convert Cycles to days*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | * 1 | * 2 | * 3 | * 4 | * 6 | * 7 | * 8 | * 10 | * 11 | * 14 | * 15 | * 17 | * 18 | * 21 | * 22 | * 24 | * 31 |
| * Days M. | * 2.47 | * 4.94 | * 7.42 | * 9.89 | * 14.83 | * 17.3 | * 19.78 | * 24.72 | * 27.19 | * 34.61 | * 37.08 | * 42.03 | * 44.5 | * 51.91 | * 54.39 | * 59.33 | * 76.64 |

* *Prove that demand is always met*

Using (T,S) policy

We are using periodic review as order intervals can be derived from order frequency for each of the items.

* Assuming demand is uniformly distributed thus the same quantity of demand repeats each time.
* Initial stock level for the 62 items are given based on lanes assigned
* Each Item has some form of order frequency which we derived cycle time from, going over a time period of 262 days. This represents order interval.
* Lead time is zero.
* At each order interval, we order up to which is the stock level based on lane assigned for each item meaning, lanes are filled.
* Backorder is allowed but customers are willing to wait
* Stock is filled to the capacity in period 1.

*Notations and Formulars*

On-hand stock S(t)

Outstanding orders O(t)

Backorders B(t)

Inventory level available units

In the code below, Shows (T,S) Policy, how demand is met over 262 time period. Reason for TS policy is that we can not exceed capacity, thus we order upto Lane capacity of each item. Also each item has its cycle time in days which was derived from order frequency M.

is the stock level, gotten from lane assigned to each items, the floor of stock level ensures that complete boxes are assigned to a lane. e.g

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| mat | erial ID b | ox ID nu | mber\_of\_lanes b\_ | not\_sorting |
| 4 | 7305673+76 | 6203059 | 2 | 396 |
| using | item 4 as an | example ag | ain |  |

That is 30 boxes in 2 lanes as a lane contains 15 boxes of item 4

Stock level derived from lanes assigned per item

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Stock level | 20 | 30 | 30 | 30 | 30 | 20 | 15 | 15 | 30 | 30 |

Periodic Inventory TS Policy

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| yt | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| qt | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 0 |
| lt | 28 | 26 | 24 | 22 | 20 | 18 | 16 | 14 | 12 | 30 | 28 |

Periodic Inventory TS Policy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |
| Average Inventory | 12.64326 | 21.96183 | 21.96183 | 22.46422 |
| Alpha Service Level | 100.00000 | 100.00000 | 100.00000 | 100.00000 |

As seen from the result, only Items 17 and 18 has an service level of 90.458% and item 57 has service level of 93.9% and the rest of the items has 100% service level.

make there service level be 100% by ordering earlier.

## [1] "box ID" "N0" "material ID"   
## [4] "demand per day" "pieces/box" "price"   
## [7] "demand\_per\_year" "box\_cost" "ordering\_cost"   
## [10] "eoq.min" "eoq.max" "b\_sorting"   
## [13] "b\_not\_sorting" "number\_of\_lanes" "Order\_frequency\_M"   
## [16] "Lot\_size\_q" "deman\_per\_day\_boxes" "cycle\_time\_in\_days"

|  |  |  |  |
| --- | --- | --- | --- |
| de | man\_per\_day\_boxes Or | der\_frequency\_M cy | cle\_time\_in\_days |
| 17 | 0.4000 | 17 | 42 |
| 18 | 0.4000 | 17 | 42 |
| 50 | 0.9375 | 7 | 17 |

Demand occurs in occurs for items not in boxes as order are done in boxes thus we store in boxes, so 0.4 is good this means the boxes are not filled.

product\_data$cycle\_time\_in\_days[c(17,18)]<-37  
product\_data$cycle\_time\_in\_days[c(50)]<-6

Periodic Inventory TS Policy

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| yt | 0.9375 | 0.9375 | 0.9375 | 0.9375 | 0.9375 | 0.9375 | 0.9375 |
| qt | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 17.0000 |
| lt | 14.0625 | 13.1250 | 12.1875 | 11.2500 | 10.3125 | 9.3750 | 15.0000 |

# Analysis

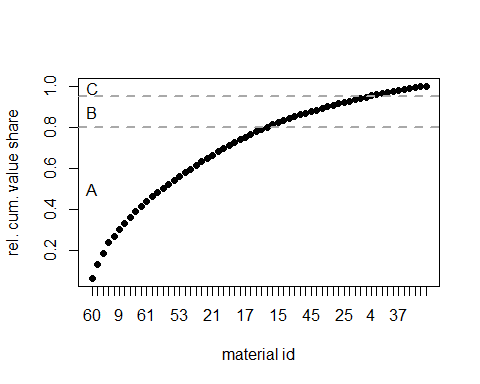
## ABC Analysis

Procedure: consumption and price of material , then 1. Calculate value share of material

1. Order materials descendingly according to value share:
2. Calculate cumulative ordered value share of each material:
3. Categorize materials according to into classes A,B and C by class limits e.g 80,95,100

ABC analysis results (values classified by boxes)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ord.material.id | price..ord.. | demand..ord.. | mat.values | cum.mat.values | rel.cum.value.shares | class |
| 60 | 256.80 | 1 | 0.07 | 0.07 | 6.7 | A |
| 36 | 252.98 | 2 | 0.07 | 0.13 | 13.2 | A |
| 50 | 215.68 | 1 | 0.06 | 0.19 | 18.8 | A |
| 57 | 196.56 | 3 | 0.05 | 0.24 | 24.0 | A |
| 52 | 127.36 | 6 | 0.03 | 0.27 | 27.3 | A |
| 9 | 121.50 | 1 | 0.03 | 0.30 | 30.4 | A |
| 40 | 116.04 | 0 | 0.03 | 0.33 | 33.4 | A |
| 2 | 114.30 | 2 | 0.03 | 0.36 | 36.4 | A |
| 56 | 110.88 | 1 | 0.03 | 0.39 | 39.3 | A |
| 59 | 95.76 | 1 | 0.02 | 0.42 | 41.8 | A |
| 61 | 87.80 | 1 | 0.02 | 0.44 | 44.0 | A |
| 62 | 87.80 | 1 | 0.02 | 0.46 | 46.3 | A |
| 10 | 76.80 | 1 | 0.02 | 0.48 | 48.3 | A |
| 34 | 76.40 | 0 | 0.02 | 0.50 | 50.3 | A |
| 32 | 75.90 | 0 | 0.02 | 0.52 | 52.3 | A |
| 16 | 75.84 | 1 | 0.02 | 0.54 | 54.2 | A |
| 53 | 73.50 | 2 | 0.02 | 0.56 | 56.2 | A |
| 6 | 72.80 | 4 | 0.02 | 0.58 | 58.0 | A |
| 39 | 68.16 | 0 | 0.02 | 0.60 | 59.8 | A |
| 1 | 67.95 | 0 | 0.02 | 0.62 | 61.6 | A |
| 7 | 66.78 | 5 | 0.02 | 0.63 | 63.3 | A |
| 33 | 63.30 | 0 | 0.02 | 0.65 | 65.0 | A |
| 21 | 63.00 | 0 | 0.02 | 0.67 | 66.6 | A |
| 51 | 62.72 | 1 | 0.02 | 0.68 | 68.2 | A |
| 55 | 60.12 | 1 | 0.02 | 0.70 | 69.8 | A |
| 27 | 59.58 | 1 | 0.02 | 0.71 | 71.3 | A |
| 19 | 55.44 | 1 | 0.01 | 0.73 | 72.8 | A |
| 13 | 51.24 | 2 | 0.01 | 0.74 | 74.1 | A |
| 17 | 49.80 | 0 | 0.01 | 0.75 | 75.4 | A |
| 22 | 47.06 | 0 | 0.01 | 0.77 | 76.6 | A |
| 18 | 46.95 | 0 | 0.01 | 0.78 | 77.8 | A |
| 54 | 46.44 | 2 | 0.01 | 0.79 | 79.1 | A |
| 35 | 45.60 | 0 | 0.01 | 0.80 | 80.2 | B |
| 11 | 45.00 | 1 | 0.01 | 0.81 | 81.4 | B |
| 15 | 44.10 | 2 | 0.01 | 0.83 | 82.6 | B |
| 30 | 37.00 | 0 | 0.01 | 0.84 | 83.5 | B |
| 42 | 35.52 | 2 | 0.01 | 0.84 | 84.4 | B |
| 31 | 32.40 | 0 | 0.01 | 0.85 | 85.3 | B |
| 29 | 32.25 | 2 | 0.01 | 0.86 | 86.1 | B |
| 28 | 31.95 | 2 | 0.01 | 0.87 | 86.9 | B |
| 45 | 30.42 | 2 | 0.01 | 0.88 | 87.7 | B |
| 44 | 30.24 | 2 | 0.01 | 0.89 | 88.5 | B |
| 20 | 30.00 | 0 | 0.01 | 0.89 | 89.3 | B |
| 46 | 29.92 | 2 | 0.01 | 0.90 | 90.1 | B |
| 47 | 28.16 | 2 | 0.01 | 0.91 | 90.8 | B |
| 23 | 27.69 | 0 | 0.01 | 0.92 | 91.5 | B |
| 25 | 27.20 | 0 | 0.01 | 0.92 | 92.2 | B |
| 24 | 24.48 | 0 | 0.01 | 0.93 | 92.9 | B |
| 14 | 24.32 | 4 | 0.01 | 0.94 | 93.5 | B |
| 3 | 24.30 | 2 | 0.01 | 0.94 | 94.1 | B |
| 12 | 24.24 | 2 | 0.01 | 0.95 | 94.8 | B |
| 4 | 24.16 | 2 | 0.01 | 0.95 | 95.4 | C |
| 5 | 24.16 | 2 | 0.01 | 0.96 | 96.0 | C |
| 38 | 22.56 | 4 | 0.01 | 0.97 | 96.6 | C |
| 41 | 21.00 | 2 | 0.01 | 0.97 | 97.1 | C |
| 43 | 20.57 | 3 | 0.01 | 0.98 | 97.7 | C |
| 37 | 20.54 | 2 | 0.01 | 0.98 | 98.2 | C |
| 48 | 17.25 | 6 | 0.00 | 0.99 | 98.7 | C |
| 8 | 17.16 | 5 | 0.00 | 0.99 | 99.1 | C |
| 26 | 13.59 | 1 | 0.00 | 0.99 | 99.5 | C |
| 58 | 12.80 | 3 | 0.00 | 1.00 | 99.8 | C |
| 49 | 7.85 | 6 | 0.00 | 1.00 | 100.0 | C |



## IQR Analysis

Categorization by life cycle - Inventory quality ratio

Aim is to minimize the risk of obselensce.

* Life cycle of product is the change of demand over time.
* Variation of material with respect to it’s importance. Typically associated with ABC analysis.
* The higher the value of materials, the potential risk of obscelence situation. This risk increase if this are compared to it’s demand overstocked.

There are 262 days working in consideration, 5 working days in a week, $=52.4 $ that is 52 weeks in a year. So we assume that current stock level is equivelent to stocks that can fit lane assignments for each item and

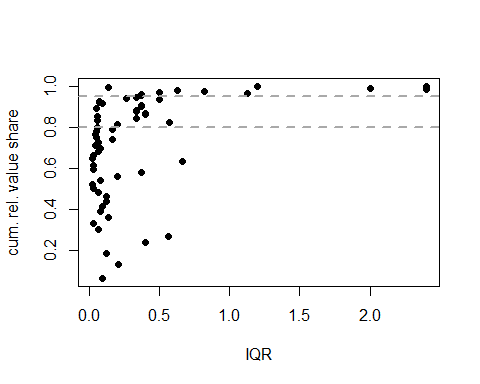
* estimated demand value per period (week).
* The current stock value.
* The accepted turnover time of material then.

We will use these thresholds : A…2 per., B..4 per. and C..6 per.

1. Calculate active inventory
2. Inventory quatlity ratio the closer to 1, the better.

Life-cycle analysis for materials 1-62

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| material.id | stock.value | to.value.act.inv | iqr | excess.inv |
| 1 | 1359.0 | 36.2 | 0.0 | 1322.8 |
| 2 | 3429.0 | 457.2 | 0.1 | 2971.8 |
| 3 | 729.0 | 194.4 | 0.3 | 534.6 |
| 4 | 724.8 | 271.8 | 0.4 | 453.0 |
| 5 | 724.8 | 271.8 | 0.4 | 453.0 |
| 6 | 1456.0 | 546.0 | 0.4 | 910.0 |
| 7 | 1001.7 | 667.8 | 0.7 | 333.9 |
| 8 | 257.4 | 514.8 | 2.0 | -257.4 |
| 9 | 3645.0 | 243.0 | 0.1 | 3402.0 |
| 10 | 2304.0 | 153.6 | 0.1 | 2150.4 |
| 11 | 900.0 | 180.0 | 0.2 | 720.0 |
| 12 | 727.2 | 242.4 | 0.3 | 484.8 |
| 13 | 1537.2 | 256.2 | 0.2 | 1281.0 |
| 14 | 729.6 | 364.8 | 0.5 | 364.8 |
| 15 | 661.5 | 378.0 | 0.6 | 283.5 |
| 16 | 2275.2 | 189.6 | 0.1 | 2085.6 |
| 17 | 747.0 | 39.8 | 0.1 | 707.2 |
| 18 | 704.2 | 37.6 | 0.1 | 666.7 |
| 19 | 1108.8 | 73.9 | 0.1 | 1034.9 |
| 20 | 660.0 | 36.0 | 0.1 | 624.0 |
| 21 | 1386.0 | 37.8 | 0.0 | 1348.2 |
| 22 | 941.2 | 43.4 | 0.0 | 897.8 |
| 23 | 553.8 | 51.1 | 0.1 | 502.7 |
| 24 | 489.6 | 36.7 | 0.1 | 452.9 |
| 25 | 544.0 | 40.8 | 0.1 | 503.2 |
| 26 | 407.7 | 54.4 | 0.1 | 353.3 |
| 27 | 1787.4 | 79.4 | 0.0 | 1708.0 |
| 28 | 639.0 | 255.6 | 0.4 | 383.4 |
| 29 | 645.0 | 258.0 | 0.4 | 387.0 |
| 30 | 740.0 | 44.4 | 0.1 | 695.6 |
| 31 | 648.0 | 38.9 | 0.1 | 609.1 |
| 32 | 1518.0 | 30.4 | 0.0 | 1487.6 |
| 33 | 1266.0 | 25.3 | 0.0 | 1240.7 |
| 34 | 1528.0 | 45.8 | 0.0 | 1482.2 |
| 35 | 912.0 | 54.7 | 0.1 | 857.3 |
| 36 | 5565.6 | 1167.6 | 0.2 | 4398.0 |
| 37 | 451.9 | 284.4 | 0.6 | 167.5 |
| 38 | 451.2 | 507.6 | 1.1 | -56.4 |
| 39 | 2044.8 | 68.2 | 0.0 | 1976.6 |
| 40 | 3481.2 | 116.0 | 0.0 | 3365.2 |
| 41 | 630.0 | 315.0 | 0.5 | 315.0 |
| 42 | 1065.6 | 355.2 | 0.3 | 710.4 |
| 43 | 411.4 | 336.6 | 0.8 | 74.8 |
| 44 | 604.8 | 201.6 | 0.3 | 403.2 |
| 45 | 608.4 | 202.8 | 0.3 | 405.6 |
| 46 | 598.4 | 224.4 | 0.4 | 374.0 |
| 47 | 563.2 | 211.2 | 0.4 | 352.0 |
| 48 | 258.8 | 621.0 | 2.4 | -362.2 |
| 49 | 117.8 | 282.6 | 2.4 | -164.9 |
| 50 | 3235.2 | 404.4 | 0.1 | 2830.8 |
| 51 | 1881.6 | 117.6 | 0.1 | 1764.0 |
| 52 | 2547.2 | 1432.8 | 0.6 | 1114.4 |
| 53 | 1470.0 | 294.0 | 0.2 | 1176.0 |
| 54 | 928.8 | 154.8 | 0.2 | 774.0 |
| 55 | 1202.4 | 100.2 | 0.1 | 1102.2 |
| 56 | 2217.6 | 184.8 | 0.1 | 2032.8 |
| 57 | 2948.4 | 1179.4 | 0.4 | 1769.0 |
| 58 | 192.0 | 230.4 | 1.2 | -38.4 |
| 59 | 2106.7 | 191.5 | 0.1 | 1915.2 |
| 60 | 5649.6 | 513.6 | 0.1 | 5136.0 |
| 61 | 1756.0 | 210.7 | 0.1 | 1545.3 |
| 62 | 1756.0 | 210.7 | 0.1 | 1545.3 |



## Cost Analysis

Total cost Matrix

|  |  |  |
| --- | --- | --- |
|  | unconstrained | constrained |
| Separate Ordering | 174363 | 206266 |
| Joint Ordering | 49943 | 268254 |

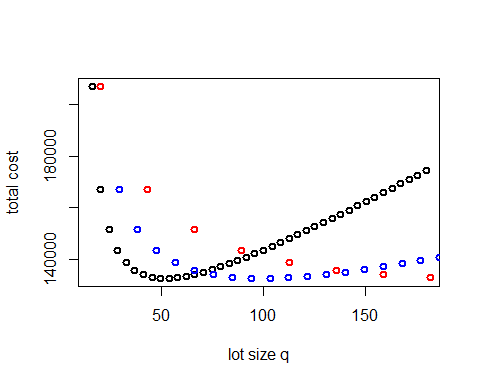
in the matrix above, it costs more to When capacity is an issue for both seperate Ordering and Joint Ordering. When non capacited ordering, there is far more cost savings than Separate ordering.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 2 57 | 36 |  | 7 60 | 6 | 53 | 2 | 50 | 61 |  |
|  | 134 | 71 | 55 | 119 | 24 | 89 | 48 | 48 | 22 | 29 |

## [1] 62

## integer(0)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | sum | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Non Constrained | 35235.00 | 180 | 381 | 825 | 802 | 802 | 652 | 787 |
| Constrained | 1140.33 | 16 | 20 | 20 | 20 | 20 | 19 | 15 |

As seen from the table, with no constraint you tend to order far more than constrained.  Without considering capacity constraint, Using some examples for illustration,for all items using the EOQ model. for Items 1 and 2, EOQ is 97 and 108 respectively without paying extra 1500.

When ordering cost of 1500 and capacity for items 1 and 2 has order quantities of 19 and 20 respectively, thus costing far more as seen in the graph above. This is largely due to lack of capacity constraint because so little is ordered, that 1500 is paided all the time order takes place for each items.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| su | m 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| No Constrained | 285 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Constrained | 544 | 24 | 4 | 4 | 4 | 4 | 2 | 1 |
| As seen above, wi | th no | capac | ity, | you g | enera | lly h | ave s | maller order frequencies across the 62 items which shows but larger order quantity with lower cost generally. |

Since we have limited capacity, we will order frequently and therefore will not take advantage of cost savings we could have gotten if we have larger capacity.