ADU = 18
DLT = 7 days
LT factor = 0.5
Var factor = 0.33

Red Zone = 84 Units
18 \times 7 \times 0.5 \times (1+0.33)

Spike Threshold = 42 Units
ADU = 2
DLT = 7 days
LT factor = 0.5
Var factor = 0.33

Red Zone = 9 Units
\[2 \times 7 \times 0.5 \times (1+0.33)\]

Spike Threshold = 4.5 Units
• When every sales order is a spike, then the system acts as a MTO: for every sales order you place a replenishment order within lead time.
• This defeats the purpose of the buffer completely
We were doing simulation with hundreds or SKU’s that have this kind of sales profile

- This is not a major issue when MOQ is >> ADU
- For all others we had to correct the issue by manually adjust the red zone and/or the threshold % factor with crazy numbers
- This was fully a trial and error exercise and extremely time consuming
- There must be a better way, a systematic way with some mathematical foundation that can be used to automate this process!

Maybe we should calculate the RED ZONE by ignoring all the Zeroes and taking only the days with sales into account?
Mathematically this is easy

If there were 41 days with sales in a year, that is on average one day every nine

\[ \frac{1}{9} \]

Then we just multiply the ADU for the Red Base calculation by 9

\[ ADU = 2 \quad \rightarrow \quad 2 \times 9 = 18 \]

ADU for the Red Zone Base calculation = 18

Red Zone = 84 Units

\[ 2 \times 7 \times 0.5 \times 9 \times (1+0.33) \]

Spike Threshold = 42 Units

This is looking great, because this is the same Red Zone and Threshold we had originally......BUT!
...It feels like this red zone just seems TOO BIG!
- Maybe we can solve the problem by just applying the factor 9 to the spike threshold

Red Zone = 9 Units

Spike Threshold = 4.5 Units \( \rightarrow 4.5 \times 9 = 42 \text{ Units} \)

Now the Red Zone is not sufficiently big to protect the buffer!
Both solutions result in a **Spike Threshold of 42 Units**, which is exactly what we want, but the red zone is either too big or too small.

Is there a middle ground?

What about splitting the factor in two components and applying each component to the Red Base and the Threshold respectively?

We used the Square Root of the factor

\[
9 = 3 \times 3
\]

**Red Zone** = \(2 \times 7 \times 0.5 \times 3 \times (1+0.33) = 28 \text{ Units}\)

**Spike Threshold** = 14 units → \(14 \times 3 = 42 \text{ Units}\)
Finally, we seem to have found an easy way to deal with these products

When we applied to our simulation we found that with these settings most buffers looked good, with very little adaption needed in some exceptional cases.

The more interspersed the sales are, the higher the factor

On the other side, when you have sales every day, or almost, then the factor $\rightarrow 1 \quad \text{Sqrt}(\rightarrow 1) = \rightarrow 1$ The factor doesn’t impact the original buffer calculation
Next step: Explore the benefit of adapting the squared root factor to the Green Zone as well?

- Benefits: Better buffering of demand variability (because of the small green zone every shipment makes the Net Flow drop in the yellow, thus triggering a replenishment order and passing the variation upstream).
- Cost: Average inventories goes up

<table>
<thead>
<tr>
<th>Order Patterns</th>
<th>Units</th>
<th>Value USD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Order</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>Min</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Max</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>Median</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td># Orders</td>
<td>35</td>
<td></td>
</tr>
</tbody>
</table>

Original Green Zone = 7

<table>
<thead>
<tr>
<th>Order Patterns</th>
<th>Units</th>
<th>Value USD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Order</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>Min</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>Max</td>
<td>76</td>
<td>76</td>
</tr>
<tr>
<td>Median</td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td># Orders</td>
<td>21</td>
<td></td>
</tr>
</tbody>
</table>

Green Zone = 7x3 = 21