metin, el yazısı, kağıt, kağıt ürünü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Optimize Class

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturulduMain METHOD

Figure Main Method-Optimize Class

In Figure 1 we created a scanner object to receive input from the user. We ensured that the application ran continuously with the while(true) loop until the user entered 0.

Inside the loop, it asks the user to enter seven values:

c: Holding cost per unit per unit time.

I: Inventory carrying cost as a percentage of unit cost.

p: Unit price (purchase price per unit).

K: Ordering cost per order.

λ: Demand rate (average number of units demanded per unit time).

μ: Production/Delivery rate (average number of units produced/delivered per unit time).

σ: Demand standard deviation (standard deviation of the demand per unit time).

Various calculations are made with the calculateOptimalValues method we created .Finally, it prints the calculated optimal values in a formatted way:

Optimal Q (order quantity)

Optimal R (reorder point)

Holding Cost

Ordering Cost

Penalty Cost

Average Time Between Orders

Proportion No Stock Out

Proportion Unmet Demand

Difference Between R and Mean Demand

calculateOptimalValues METHOD

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu In figure 2 calculates various inventory management optimal order quantity and reorder point. This calculation process is done iteratively and continues until a certain tolerance or maximum number of iterations is reached.

Figure calculateOptimalValues - Optimize Class

*h*: Holding cost per unit.

*Q0*: Initial order quantity.

*F\_R0*: Service level function.

*current\_z*: z-value from normal distribution.

*prev\_R*: First reorder point.

ekran görüntüsü, metin, multimedya yazılımı, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu The cycle continues until the maximum number of iterations or until a certain tolerance is reached. The loss function value is found for current\_z. The updated reorder point is calculated. The standard deviation multiplier is calculated with the loss function value. The new order quantity is calculated with the economic order quantity formula. If the difference between Q and Q0 is less than the tolerance, the cycle is terminated. If the difference between R and prev\_R is less than the tolerance and this is not the first iteration, the loop is terminated. The service level function is updated. Q0, prev\_R and current\_z are updated with the newly calculated values.

Figure

Optimal Q: optimalValues[0]

Optimal R: optimalValues[1]

Holding Cost: optimalValues[2]

Ordering Cost: optimalValues[3]

Penalty Cost: optimalValues[4]

Average Order Range: optimalValues[5]

Stockout Rate: optimalValues[6]

Unmet Demand Rate: optimalValues[7]

R and Average Demand Difference: optimalValues[8]

ReadZtable Class

This piece of code defines the Read table class. This class is used to read z and L values from a table, which are used in inventory management calculations.

calculateZ METHOD

Figure calculateZ Method-ReadZtable Class

In figüre 4 it finds the closest z-value from a file (TSV format) for a given probability value. To find the z-value closest to an entered probability value.metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturulduFirst we defined the variables;

filePath: The path to the file containing the Z-values and corresponding probabilities.

line: Keeps every line read from the file.

delimiter: It is the tab character used to separate the columns in the file.

min\_z: Keeps the closest z-value.

min\_distance: Keeps the smallest distance. It is initially set to a very large value so that a smaller value can be found.

First, the method determines a file path (zChart.tsv) and uses BufferedReader to read that file line by line. Since the first line contains header information, it skips this line. It then splits each line by the tab character and compares the probability value in that line with the entered probability value. The comparison is made by calculating the difference between two values. If this difference is smaller than the previous smallest difference, then the corresponding z-value and distance are updated. Thus, at the end of the loop, the z-value closest to the entered probability value is found. This z-value is returned as the return value of the method.

findL METHOD

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure findL Method- readZtable Class

This method finds the corresponding L(z) value from a file (TSV format) for a given z-value. Finds the closest L(z) value for an entered z-value. First we defined the variables;

filePath: The path to the file containing L(z) values.

line: Keeps every line read from the file.

delimiter: It is the tab character used to separate the columns in the file.

min\_Lz: Keeps the closest L(z) value.

min\_distance: Keeps the smallest distance. It is initially set to a large value so that a smaller value can be found.

The method works by reading a file containing z-values and corresponding L(z) values line by line. The first line is skipped because it contains header information. The z-value in each row is then compared to the entered z-value and the difference between them is calculated. If this difference is less than the smallest difference, the corresponding L(z) value and distance are updated. This process continues until the end of the file, and as a result, the L(z) value closest to the entered z-value is found. The method completes the operation by returning this L(z) value.

gitHub Repository Link: https://github.com/beyzackaya/ENM320-Production-and-Operations-Planning-II-B-M208-Computer-Programming-IV-Term-Project.git