Application of MSC in Inventory Management of Perishable Products

Alireza Sheikh-Zadeh, Ph.D.

The decision model for managing the inventory of perishable products is known as the newsvendor problem or newsboy problem. This model is based on an analogy with the situation faced by a newspaper vendor who must decide how many copies of the day's paper to stock in the face of uncertain demand and knowing that unsold copies will be worthless at the end of the day. Monte-Carlo Simulation is one of the solution methods for solving the newsvendor problems.

Newsvendor Model: Single-Period Purchase Decisions

A street newsvendor sells daily newspapers and must decide how many copies to purchase at the beginning of the day. Purchasing too few items results in a lost opportunity to increase profits, but purchasing too many items results in a loss since the excess must be discarded (or sold at a lower price).

To better understand this problem, we first develop a general model and then illustrate it with an example.

Here we assume that each item costs $\$C$ and can be sold for $\$R$. At the end of the period, any unsold items can be disposed of at $\$S$ each (the salvage value). Therefore, it makes sense to assume that $R > C > S$.

We also assume D is the number of units demanded daily, and Q is the quantity purchased at the beginning of the day. Note that D is an uncontrollable input, whereas Q is a decision variable. If demand is known, then the optimal decision is obvious: Choose $Q = D$. However, since D is unknown in advance, we face the risk of over-stocking or under-stocking. If $Q < D$, then we lose the opportunity of realizing additional profit (because the selling price is more than cost: $R>C$), and if $Q>D$, we incur a loss for disposing of the excess inventory (because the salvage value is less than the purchase cost: $C>S$).

Notice that we cannot sell more than the minimum of the actual demand and the amount purchased. Thus, the quantity sold at the regular price is the minimum of D and Q. Also, the excess (unsold) quantity can be negative; therefore, the excess quantity is the maximum of 0 and $Q - D$.

Using this information, we calculate the daily net profit as:

$$\text{Net profit} = R \times min(Q, D) + S \times max(0, Q - D) - C \times Q$$

In reality, demand D is uncertain.

Example: A Single-Period Purchase Decision Model

Suppose that a small candy store makes Valentine's Day gift boxes that cost \$12.00 and sell for \$18.00. In the past, at least 40 boxes have been sold by Valentine's Day, but the actual amount is uncertain, and in the past, the owner has often run short or made too many. After the holiday, any unsold boxes are discounted by 50% and are eventually sold.

The net profit can be calculated for any values of Q and D:

net profit = \$18.00 × min{D,Q} + \$9.00 × max{0, $Q - D$} - \$12.00 × Q

```
# Data
R = 18   # Selling price
C = 12   # Cost
S = 9    # Discount Price

# Model
netProfitFun = function(D, Q, R, S, C){
  R*min(D,Q) + S * max(0, Q-D) - C*Q
}

# If D is given, the best solution Q = D.
```

Suppose that the store owner kept records for the past 20 years on the number of boxes sold. The historical candy sales average is 44.05.

```
histDemand = read.csv("histDemand.csv") # File is att
ached on Blackboard

histDemand$Sales

##  [1] 42 45 40 46 43 43 46 42 44 43 47 41 41 45 51
43 45 42 44 48

mean(histDemand$Sales)
```

```
## [1] 44.05
```

As the average demand is ≈ 44, the candy store can decide to make $Q = 44$ gift boxes to satisfy demand. In bellow, we show that this is not the best strategy.

The Flaw of Averages

The model output using the average value of the input is not necessarily equal to the average value of the outputs when evaluated with each input value.

Using 44 for demand and Q, the model predicts a profit of \$264.00.

```
# Based on average historical data (we round it down
to the nearest integer)
mean_D = 44

# Rationally, we make 44 boxes
Q = 44

netProfitFun(mean_D, Q, R, S, C)

## [1] 264
```

However, if we construct a data table to evaluate the profit for each historical value, the average profit is only \$255.00.

```
# Assuming we make 44 boxes
Q = 44

hist_D = histDemand$Sales


profitVector <- c()

for (i in 1:length(hist_D)) {
  profitVector[i] = netProfitFun(hist_D[i], Q, R, S,
C)
}

# Or you can use sapply function instead of for-loop
# profitVector = sapply(hist_D, netProfitFun, Q, R, S
, C)

mean(profitVector)

## [1] 255
```

```
# Which is less than $264
```

The evaluation of a model output using the average value of the input (profit =264) is not necessarily equal to the average value of the outputs when evaluated with each of the input values (profit =255). Why? In the newsvendor example, the quantity sold is limited to the smaller of the demand and purchase quantity (min(*D*, *Q*)), so even when demand exceeds the purchased quantity, the profit is limited.)

Simulating the Newsvendor Model Using Resampling (from historical data)

Generate candy sales by resampling from the 20 historical values.

```
set.seed(123)

num_sim = 10000


# Here, we assume that the data generation process fo
llows the empirical distribution of the historical da
ta

sim_D <- sample(hist_D, num_sim, replace = TRUE) # re
sampling

profit_sim <- c()

for (i in 1:num_sim) {
  profit_sim[i] = netProfitFun(sim_D[i], Q, R, S, C)
}

mean(profit_sim)

## [1] 255.0594

# This is the expected profit given the historical da
ta empirical distribution (this method of simulation
is called Bootstrapping).
```

Question: In the Newsvendor Model, suggest the best purchase quantity Q using resampling from historical data.

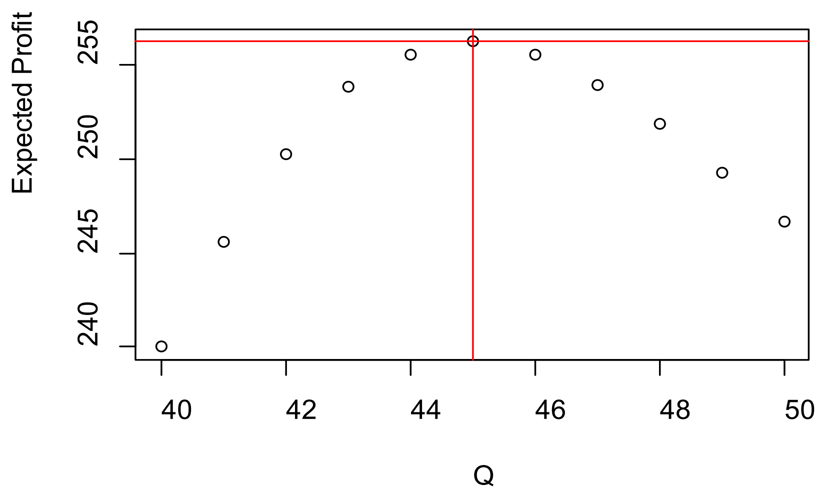
```
# Recall Data Again
R = 18  # Selling price
```

```r
C = 12  # Cost
S = 9   # Discount Price

# Q = 44 is suggested. Is it the best? If not, what Q
do you suggest?

num_sim = 1000


# We evaluate a range of possible Q values
Qrange = 40:50

sim_D <- sample(hist_D, num_sim, replace = TRUE) # re
sampling


# For each possible Q we simulate num_sim times, so f
or recording simulation values we need a matrix (Qs a
re for each column)
profitMatrix <- matrix(nrow = num_sim, ncol = length(
Qrange))

j = 0
for (Q in Qrange) {
  j = j+1
  for (i in 1:num_sim) {
    profitMatrix[i, j] = netProfitFun(sim_D[i], Q, R,
S, C)
  }
}

# Expected (average) profit for each Q
profit_for_each_Q <- colMeans(profitMatrix)

cbind(Qrange, profit_for_each_Q)
```

```
      Qrange profit_for_each_Q
 [1,]     40           240.000
 [2,]     41           245.559
 [3,]     42           250.290
 [4,]     43           253.797
 [5,]     44           255.531
 [6,]     45           256.221
 [7,]     46           255.543
 [8,]     47           253.911
 [9,]     48           251.829
[10,]     49           249.243
[11,]     50           246.657
```

```r
plot(Qrange, profit_for_each_Q, ylab = "Expected Prof
it", xlab = "Q")
```

```
best_Q_index <- which.max(profit_for_each_Q)

abline(v = Qrange[best_Q_index], h = profit_for_each_
Q[best_Q_index], col = "red")
```




```
# Q* = 45, The optimal Q is 45
```

Resampling from empirical data has some drawbacks:

- The empirical data may not represent the true underlying population adequately because of sampling error.
- Using an empirical distribution precludes sampling values outside the actual data range. It is usually advisable to fit a distribution (e.g., normal, Poisson, uniform, etc.) and use it to simulate instances for the uncertain variable.

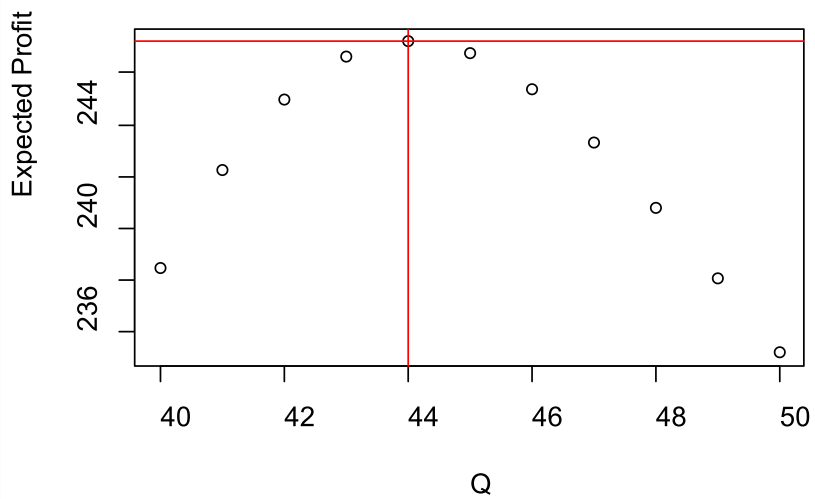
Question: In the Newsvendor Model, suggest the best purchase quantity Q using a truncated normal distribution with a mean of 42, the standard deviation of 3.4, and the lower bound of 38.

```
# Recall Data Again
R = 18   # Selling price
C = 12   # Cost
S = 9    # Discount Price

# Q = 44 is suggested. Is it the best? If not, what Q
```

```r
do you suggest?

num_sim = 1000


# We evaluate a range of possible Q values
Qrange = 40:50


# Here, we need to change the data generating process
to truncnorm distribution

Library(truncnorm)
sim_D <- rtruncnorm(num_sim, a = 38, mean = 42, sd =
3.4) # resampling

profitMatrix <- matrix(nrow = num_sim, ncol = length(
Qrange))

j = 0
for (Q in Qrange) {
  j = j+1
  for (i in 1:num_sim) {
    profitMatrix[i, j] = netProfitFun(sim_D[i], Q, R,
S, C)
  }
}

# Expected (average) profit for each Q
profit_for_each_Q <- colMeans(profitMatrix)

cbind(Qrange, profit_for_each_Q)
```

```
      Qrange profit_for_each_Q
 [1,]     40          238.6202
 [2,]     41          242.5041
 [3,]     42          245.2420
 [4,]     43          246.7275
 [5,]     44          246.9988
 [6,]     45          246.2963
 [7,]     46          244.8826
 [8,]     47          242.9046
 [9,]     48          240.4510
[10,]     49          237.6915
[11,]     50          234.7834
```

```r
best_Q_index <- which.max(profit_for_each_Q)

abline(v = Qrange[best_Q_index], h = profit_for_each_
Q[best_Q_index], col = "red")
```




```
# Q* = 44, The optimal Q is 44
```