A Prescriptive Model for Strategic Decision-making, An Inventory Management Decision Model

Yinan Zhou

16th June 2024

Table of Contents

Introduction 3
Part1 - Excel Solution 4
Part 1 – R Solution 7
Part 2 11
References 22
R Code 23

Introduction

Inventories represent a considerable investment for every organization; thus, it is important that they be managed well. Excess inventories can indicate poor financial and operational management. On the other hand, not having inventory when it is needed can also result in business failure. The two basic inventory decisions that managers face are how much to order or produce for additional inventory, and when to order or produce it to minimize total inventory cost, which consists of the cost of holding inventory and the cost of ordering it from the supplier.

Holding costs, or carrying costs, represent costs associated with maintaining inventory. These costs include interest incurred or the opportunity cost of having capital tied up in inventories; storage costs such as insurance, taxes, rental fees, utilities, and other maintenance costs of storage space; warehousing or storage operation costs, including handling, recordkeeping, information processing, and actual physical inventory expenses; and costs associated with deterioration, shrinkage, obsolescence, and damage. Total holding costs are dependent on how many items are stored and for how long they are stored. Therefore, holding costs are expressed in terms of dollars associated with carrying one unit of inventory for one unit of time. Ordering costs represent costs associated with replenishing inventories. These costs are not dependent on how many items are ordered at a time, but on the number of orders that are prepared.

Ordering costs include overhead, clerical work, data processing, and other expenses that are incurred in searching for supply sources, as well as costs associated with purchasing, expediting, transporting, receiving, and inspecting. It is typical to assume that the ordering cost is constant and is expressed in terms of dollars per order.

For a manufacturing company that you are consulting for, managers are unsure about making inventory decisions associated with a key engine component. The annual demand is estimated to be 15,000 units and is assumed to be constant throughout the year. Each unit costs $80. The company’s accounting department estimates that its opportunity cost for holding this item in stock for one year is 18% of the unit value. Each order placed with the supplier costs $220. The company’s policy to order whenever the inventory level reaches a predetermined reorder point that provides sufficient stock to meet demand until the supplier’s order can be shipped and received; and then to order twice as many units.

Part1 – Excel Solution

Task:

As a consultant, your task is to develop and implement a decision model to help the company make the best inventory decisions.

1. Define the data, uncontrollable inputs, model parameters, and the decision variables that influence the total inventory cost.

Annual Demand (D)	15000	Uncontrolable Input
Unit cost (C)	$80.00	Parameter
Carrying / Holding cost rate (H)	18%	Parameter
Order cost (S)	$220.00	Parameter
Inventory Level Order Q	341	Decision Variable
Order Quantity (2Q)	682	Decision Variable
Number of Times to Order	22	Decision Variable
Annual Ordering Cost = S * N	$4,840.00	
Annual Holding Cost = Q * C * H	$4,909.09	
Total Cost = AOC + AHC	$9,749.09	Objective: Minimize

Uncontrollable input: Annual Demand.

Parameters: Unit Cost, Carrying Cost Rate, Order Cost.

Decision Variables: Inventory Level Order, Order Quantity, Number of Times to Order.

2. Develop mathematical functions that compute the annual ordering cost and annual holding cost based on average inventory held throughout the year and use them to develop a mathematical model for the total inventory cost.

Annual Ordering Cost (AOC) = Order Cost (S) * Number of Times to Order (N)

Annual Holding Cost (AHC) = Inventory Quantity (Q) * Unit Cost (C) * Holding Cost Rate (H)

Total Inventory Cost = AOC + AHC

3. Implement your model on an Excel spreadsheet.

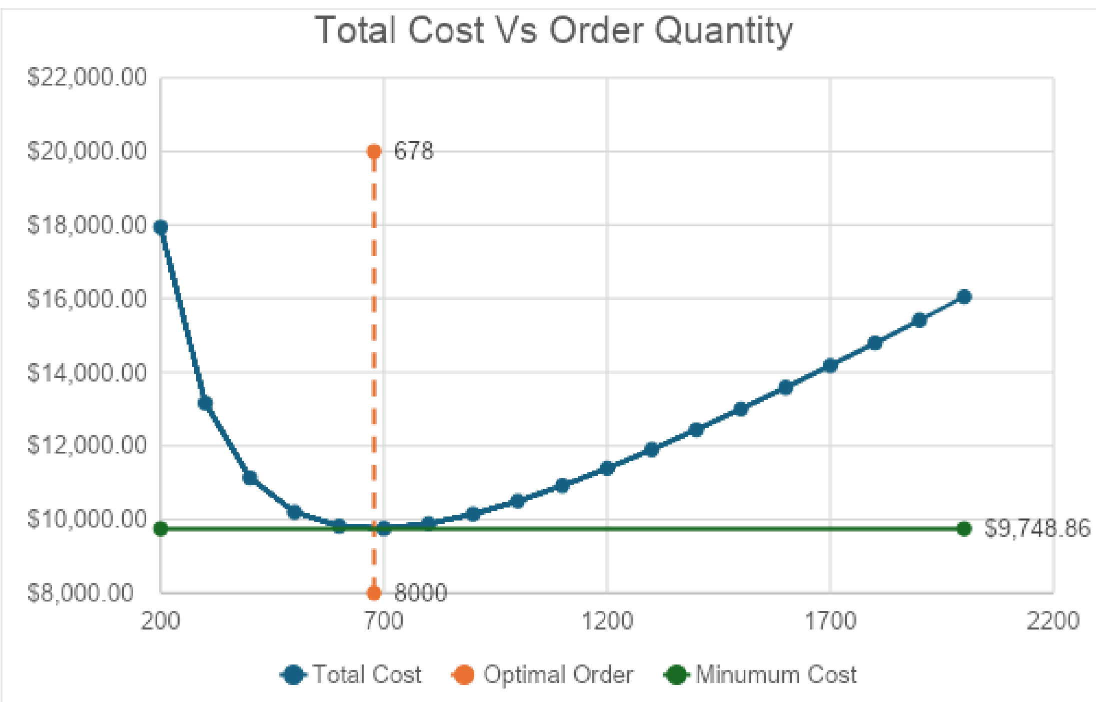
Annual Ordering Cost = S * N	$4,840.00	
Annual Holding Cost = Q * C * H	$4,909.09	
Total Cost = AOC + AHC	$9,749.09	Objective: Minimize

4. Use data tables to find an approximate order quantity that results in the smallest total cost.

Analytic Analysis:	
Total Cost = AOC + AHC	
AOC = 220 * (15000 / 2Q) = 1650000 / Q	
AHC = Q * 80 * 0.18 = 14.4*Q	
Total Cost = 1650000 / Q + 14.4*Q	
Find the Minimal	
C'(Q) --> 0 = 14.4 - 1650000 / Q^2	Differenciate = 0
Q^2 = 1650000 / 14.4	**Optimal Q:**
Q = Sqrt(1650000 / 14.4) = 339	**339**
Theoretical Total Cost	**$9,748.86**

We first calculate the AOC and AHC with unknown parameters Q, we aim to find the smallest total cost, then we added up AOC and AHC to get the total cost function: 1650000/Q +14.4*Q, then, we differentiate the total cost and set it to zero to find the smallest value of Q. The theoretical optimal Q is 339 and total cost is $9748.86

5. Plot the Total Cost versus the Order Quantity



This is the graph of Total Cost versus the Order Quantity, the blue line represents the total cost, the orange dash line represents the optimal order quantity which is 678, and the red solid line represents the minimum total cost $9748.86.

6. Use the Excel Solver to verify your result of part 4 above; that is, find the order quantity which would yield a minimum total cost.

Annual Demand (D)	15000	Uncontrolable Input
Unit cost (C)	$80.00	Parameter
Carrying / Holding cost rate (H)	18%	Parameter
Order cost (S)	$220.00	Parameter
Inventory Level Order Q	341	Decision Variable
Order Quantity (2Q)	682	Decision Variable
Number of Times to Order	22	Decision Variable
Annual Ordering Cost = S * N	$4,840.00	
Annual Holding Cost = Q * C * H	$4,909.09	
Total Cost = AOC + AHC	$9,749.09	Objective: Minimize

These are the outcomes after I run the solver in Excel, which is very close to the theoretical order quantity and total cost.

Theoretical Minimum Order Quantity: 678.

Theoretical Minimum Total Cost: $9748.86.

Solver Minimum Order Quantity: 682.

Solver Minimum Total Cost: $9749.09.

7. Conduct what-if analyses by using two-way tables in Excel to study the sensitivity of total cost to changes in the model parameters.

We conducted four two-way tables, which are Demand vs Holding Cost Rate, Demand vs Unit Cost, Demand vs Order Cost and Unit Cost vs Order Cost.

Demand Range: 12000 to 18000

Holding Cost Rate Range: 15% to 21%

Unit Cost Range: $50 to $110

Order Cost Range: $190 to $250

The two-way tables help us to find the changes of the total cost after the changes of parameters value. I also use the color conditional formatting to distinguish the values. The dark red represents the smallest total cost in the table.

Two-way Table								
$9,748.86	15%	16%	17%	18%	19%	20%	21%	Holding Cost Rate
12000	$7,961.81	$8,233.01	$8,504.21	$8,775.41	$9,046.61	$9,317.81	$9,589.01	
13000	$8,286.29	$8,557.49	$8,828.69	$9,099.89	$9,371.09	$9,642.29	$9,913.49	
14000	$8,610.77	$8,881.97	$9,153.17	$9,424.37	$9,695.57	$9,966.77	$10,237.97	
15000	$8,935.26	$9,206.46	$9,477.66	$9,748.86	$10,020.06	$10,291.26	$10,562.46	
16000	$9,259.74	$9,530.94	$9,802.14	$10,073.34	$10,344.54	$10,615.74	$10,886.94	
17000	$9,584.22	$9,855.42	$10,126.62	$10,397.82	$10,669.02	$10,940.22	$11,211.42	
18000	$9,908.71	$10,179.91	$10,451.11	$10,722.31	$10,993.51	$11,264.71	$11,535.91	
Demand								
$9,748.86	$50.00	$60.00	$70.00	$80.00	$90.00	$100.00	$110.00	Unit Cost
12000	$6,944.81	$7,555.01	$8,165.21	$8,775.41	$9,385.61	$9,995.81	$10,606.01	
13000	$7,269.29	$7,879.49	$8,489.69	$9,099.89	$9,710.09	$10,320.29	$10,930.49	
14000	$7,593.77	$8,203.97	$8,814.17	$9,424.37	$10,034.57	$10,644.77	$11,254.97	
15000	$7,918.26	$8,528.46	$9,138.66	$9,748.86	$10,359.06	$10,969.26	$11,579.46	
16000	$8,242.74	$8,852.94	$9,463.14	$10,073.34	$10,683.54	$11,293.74	$11,903.94	
17000	$8,567.22	$9,177.42	$9,787.62	$10,397.82	$11,008.02	$11,618.22	$12,228.42	
18000	$8,891.71	$9,501.91	$10,112.11	$10,722.31	$11,332.51	$11,942.71	$12,552.91	
Demand								
$9,748.86	$190.00	$200.00	$210.00	$220.00	$230.00	$240.00	$250.00	Order Cost
12000	$8,244.43	$8,421.42	$8,598.41	$8,775.41	$8,952.40	$9,129.39	$9,306.38	
13000	$8,524.67	$8,716.41	$8,908.15	$9,099.89	$9,291.63	$9,483.37	$9,675.11	
14000	$8,804.90	$9,011.39	$9,217.88	$9,424.37	$9,630.86	$9,837.35	$10,043.84	
15000	$9,085.14	$9,306.38	$9,527.62	$9,748.86	$9,970.10	$10,191.33	$10,412.57	
16000	$9,365.38	$9,601.36	$9,837.35	$10,073.34	$10,309.33	$10,545.32	$10,781.31	
17000	$9,645.61	$9,896.35	$10,147.09	$10,397.82	$10,648.56	$10,899.30	$11,150.04	
18000	$9,925.85	$10,191.33	$10,456.82	$10,722.31	$10,987.79	$11,253.28	$11,518.77	
Unit Cost								
$9,748.86	$190.00	$200.00	$210.00	$220.00	$230.00	$240.00	$250.00	Order Cost
$50.00	$7,254.54	$7,475.78	$7,697.02	$7,918.26	$8,139.50	$8,360.73	$8,581.97	
$60.00	$7,864.74	$8,085.98	$8,307.22	$8,528.46	$8,749.70	$8,970.93	$9,192.17	
$70.00	$8,474.94	$8,696.18	$8,917.42	$9,138.66	$9,359.90	$9,581.13	$9,802.37	
$80.00	$9,085.14	$9,306.38	$9,527.62	$9,748.86	$9,970.10	$10,191.33	$10,412.57	
$90.00	$9,695.34	$9,916.58	$10,137.82	$10,359.06	$10,580.30	$10,801.53	$11,022.77	
$100.00	$10,305.54	$10,526.78	$10,748.02	$10,969.26	$11,190.50	$11,411.73	$11,632.97	
$110.00	$10,915.74	$11,136.98	$11,358.22	$11,579.46	$11,800.70	$12,021.93	$12,243.17	

Part 1 – R Solution

Task:

As a consultant, your task is to develop and implement a decision model to help the company make the best inventory decisions.

1. Define the Data, Uncontrollable Inputs, Model Parameters, and Decision Variables

- **Data and Uncontrollable Inputs:**

- Annual demand (D)

- **Model Parameters:**

- Unit cost (C): $80
- Holding cost rate (h): 18% of unit cost per year
- Ordering cost (S): $220 per order

- **Decision Variables:**

- Order quantity (Q): The number of units to order each time an order is placed.

- Reorder Point (Q/2)

- Number of Times to Order

```
> # Define constants
> D <- 15000      # Annual demand
> C <- 80         # Cost per unit
> h <- 0.18       # Holding cost rate
> S <- 220        # Ordering cost per time
```

2. Develop Mathematical Functions for Inventory Costs

- **Annual Ordering Cost:**
  Ordering Cost = (D/Q) × S

- **Annual Holding Cost:**
  Holding Cost = (Q/2) × C × h

- **Total Inventory Cost:**
  Total Cost = Ordering Cost + Holding Cost = (D/Q) × S + (Q/2) × C × h

```
> # Define a function to calculate total cost
> total_cost <- function(Q) {
+   ordering_cost <- (D / Q) * S
+   holding_cost <- (Q / 2) * C * h
+   return(ordering_cost + holding_cost)
+ }
```

3. Implement the Model in R

```
> # Use the optimize function to find the optimal order quantity
> opt_result <- optimize(total_cost, c(1, 2000))
>
> # Extract the optimal order quantity and round it up
> optimal_Q <- opt_result$minimum
> # Calculate the minimum total cost
> optimal_cost <- total_cost(optimal_Q)
>
> # Print the results
> cat("Optimal Order Quantity:", optimal_Q, "\n")
Optimal Order Quantity: 677.0032
> cat("Minimum Total Cost:", optimal_cost, "\n")
Minimum Total Cost: 9748.846
```

R results:

- **Optimal Order Quantity:** 677.0032 units

- **Minimum Total Cost:** $9,748.846

4. Create a Data Table to Find the Order Quantity with the Minimum Total Cost

Since the order quantity must be an integer, we used a data table to find the exact order quantity that results in the smallest total cost.

We generated a sequence of order quantities and calculated the total cost for each quantity, and then stored these in a data table.

```
> # Generate a sequence of order quantities
> Q_values <- seq(500, 1000, by=1)
>
> # Calculate the total cost for each order quantity
> costs <- sapply(Q_values, total_cost)
>
> # Create a data table
> dt <- data.table(Order_Quantity = Q_values, Total_Cost = costs)
>
> # Print the first few rows of the data table
> head(dt)
   Order_Quantity Total_Cost
            <num>      <num>
1:            500   10200.00
2:            501   10194.03
3:            502   10188.11
4:            503   10182.24
5:            504   10176.42
6:            505   10170.65
```

Next, we used the “which.min” function to find the index of the minimum total cost and then use this index to find the corresponding order quantity.

```
> # Find the order quantity corresponding to the minimum total cost
> optimal_Q <- dt$Order_Quantity[min_cost_index]
> optimal_cost <- dt$Total_Cost[min_cost_index]
>
> # Print the optimal order quantity and the corresponding total cost
> cat("Optimal Order Quantity:", optimal_Q, "\n")
Optimal Order Quantity: 677
> cat("Minimum Total Cost:", optimal_cost, "\n")
Minimum Total Cost: 9748.846
```

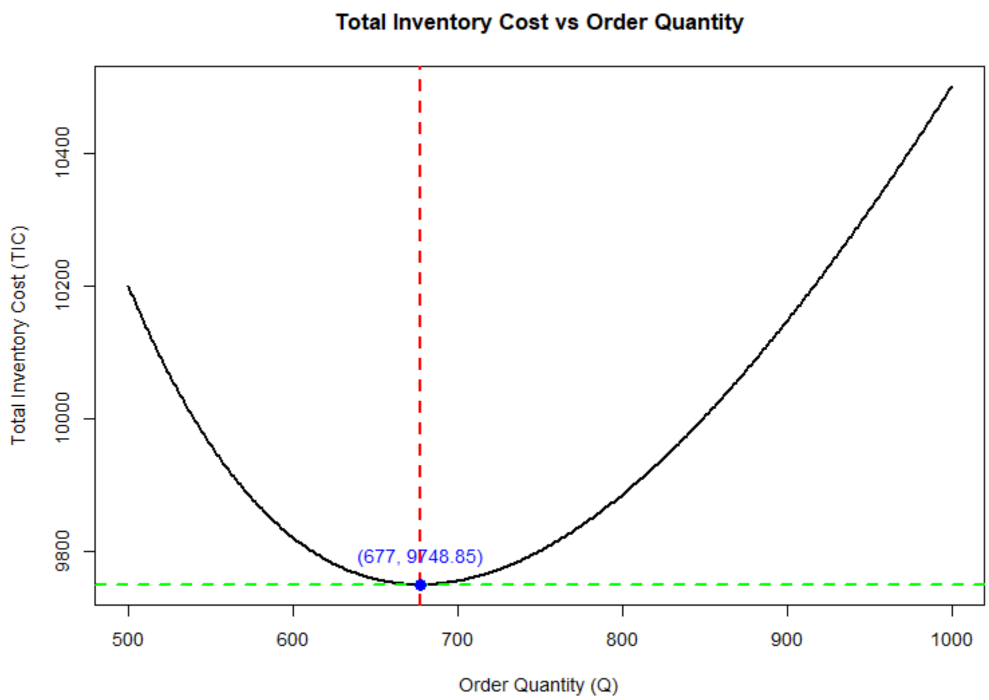
R results:

- **Optimal Order Quantity:** 677 units
- **Minimum Total Cost:** $9,748.846

5. Plot Total Cost vs. Order Quantity

The plot of Total Cost versus Order Quantity helps to understand how the total cost varies with different order quantities. It shows a clear minimum point at the optimal order quantity. The vertical red dashed line indicates the optimal order quantity, while the horizontal green dashed line represents the minimum total cost. The blue point marks the intersection point of the optimal order quantity and the minimum total cost.

```
> # Plot Total Cost vs Order Quantity
> plot(dt$Order_Quantity, dt$Total_Cost, type = "l", col = "black", lwd = 2,
+      xlab = "Order Quantity (Q)", ylab = "Total Inventory Cost (TIC)",
+      main = "Total Inventory Cost vs Order Quantity")
>
> # Add ablines for optimal Q and optimal cost
> abline(v = optimal_Q, col = "red", lwd = 2, lty = 2)
> abline(h = optimal_cost, col = "green", lwd = 2, lty = 2)
>
> # Mark the intersection point
> points(optimal_Q, optimal_cost, col = "blue", pch = 19, cex = 1.5)
> text(optimal_Q, optimal_cost, labels = paste("(", optimal_Q, ", ", round(optimal_cost, 2), ")", sep = ""),
+      pos = 3, offset = 1, col = "blue")
```



6. Recommendations for the Vice President of Operations

- **Implementation of Optimal Order Quantity**

  The company should implement an order quantity of 677 units to achieve the lowest possible total inventory cost of $9,748.85. This implementation should be monitored and adjusted periodically based on actual demand and cost fluctuations.

- **Regular Review and Adjustment**

  Given that demand, costs, and other factors can change, it is advisable to review and adjust the order quantity at regular intervals (e.g., quarterly or biannually). This ensures that the inventory management strategy remains aligned with current business conditions. (Pienaar, n.d.)

- **Use of Technology and Tools**

  Consider using inventory management software that can automatically calculate optimal order quantities and trigger reorder points based on real-time data. This will streamline operations and reduce the risk of stockouts or overstocking. (Bennett, 2024)

- **Training and Communication**

Ensure that relevant staff are trained on the new inventory management procedures and understand the rationale behind the optimal order quantity. Clear communication will help in the smooth implementation and adherence to the new strategy.

- **Continuous Improvement**

  Encourage a culture of continuous improvement where the inventory management process is regularly analyzed for potential enhancements. Gathering feedback from operational staff and monitoring key performance indicators (KPIs) will be crucial.

Part2

Task:

Assume that all problem parameters have the same values as those in part I, but that the annual demand has a triangular probability distribution between 13000 and 17000 units with a mode of 15000 units. Perform a simulation consisting of 1000 occurrences and calculate the minimum total cost for each occurrence. Next, use the results of your simulation to estimate the expected minimum total cost, optimal order quantity, and the expected annual number of orders. Additionally, the report identifies the best-fitting probability distributions for these estimates and validates their appropriateness.

1. Preliminary Work

Demand Simulation

A custom function “rtriangular” was defined to generate random variables from a triangular distribution. This function was then used to generate n random demand values.

```
> # Define parameters
> set.seed(123)  # Set seed for reproducibility
> n <- 1000 # n_simulations
>
> a <- 13000 # min_demand
> b <- 17000 # max_demand
> c <- 15000 # mode_demand
>
> # Function to generate random variates from a triangular distribution
> rtriangular <- function(n, a, b, c) {
+   u <- runif(n)
+   return(ifelse(u < (c - a) / (b - a),
+                 a + sqrt(u * (b - a) * (c - a)),
+                 b - sqrt((1 - u) * (b - a) * (b - c))))
+ }
>
> # Generate demand data
> D <- rtriangular(n, a, b, c)
```

Total Cost Calculation:

The function “total_cost” was defined to compute the total cost by summing the ordering cost and holding cost.

```
> # Define constants
> C <- 80          # Cost per unit
> h <- 0.18        # Holding cost rate
> S <- 220         # Ordering cost per time
>
> # Define a function to calculate total cost
> total_cost <- function(Q, D, S, C, h) {
+   ordering_cost <- (D / Q) * S
+   holding_cost <- (Q / 2) * C * h
+   return(ordering_cost + holding_cost)
+ }
```

Simulation Loop

The simulation loop iterates over each simulated demand value to determine the optimal order quantity and the minimum total cost.

For each demand value, the "optimize" function finds the order quantity that minimizes the total cost, which is then rounded up to the nearest integer. The optimal order quantity, minimum total cost, and the number of orders are stored in vectors.

```
> # Initialize vectors to store results
> optimal_Q <- numeric(n)
> min_total_cost <- numeric(n)
> number_of_orders <- numeric(n)
>
> # Iterate through demand vector D, calculating the optimal order quantity and minimum total cost for each demand
> for (i in 1:n) {
+   # Current demand
+   current_D <- D[i]
+   # Define the total cost function for the current demand
+   current_total_cost <- function(Q) {
+     return(total_cost(Q, current_D, S, C, h))
+   }
+   # Use the optimize function to find the optimal order quantity
+   opt_result <- optimize(current_total_cost, c(1, 2000))
+   # Extract the optimal order quantity and round it up
+   optimal_Q[i] <- ceiling(opt_result$minimum)
+   # Calculate the minimum total cost for the optimal order quantity
+   min_total_cost[i] <- current_total_cost(optimal_Q[i])
+   # Calculate the number of orders based on current demand and optimal order quantity
+   number_of_orders[i] <- current_D / optimal_Q[i]
+ }
```

2. Solutions

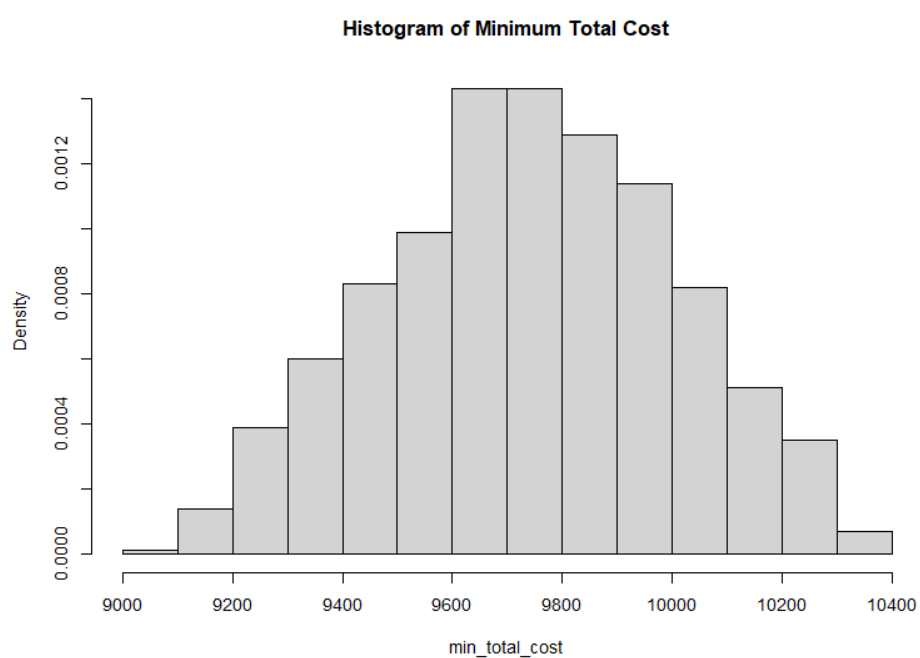
(i) Estimate the expected minimum total cost by constructing a 95% confidence interval for it and determine the probability distribution that best fits its distribution. Verify the validity of your choice.

We calculated the mean and standard deviation of the simulated minimum total costs ("min_total_cost") and computed a 95% confidence interval using standard error and a z-score.

```
> # (i) Estimate expected minimum total cost and its 95% confidence interval
> mean_min_total_cost <- mean(min_total_cost)
> sd_min_total_cost <- sd(min_total_cost)
> n_obs <- length(min_total_cost)
> se_min_total_cost <- sd_min_total_cost / sqrt(n_obs)
> z_score <- qnorm(0.975)  # For 95% confidence interval
> ci_min_total_cost <- c(mean_min_total_cost - z_score * se_min_total_cost,
+                        mean_min_total_cost + z_score * se_min_total_cost)
>
> # Output the results
> cat("Estimated Expected Minimum Total Cost:\n")
Estimated Expected Minimum Total Cost:
> cat("Mean:", mean_min_total_cost, "\n")
Mean: 9742.772
> cat("95% Confidence Interval:", ci_min_total_cost, "\n\n")
95% Confidence Interval: 9726.393 9759.151
```

Then, we created a histogram to visualize the distribution of minimum total costs and computed skewness and kurtosis for further insights into the distribution's shape.

```
> # Histogram plot to visualize the distribution
> hist(min_total_cost, freq = FALSE, main = "Histogram of Minimum Total Cost")
>
> library(fBasics)
> skewness(min_total_cost)
[1] -0.05346987
attr(,"method")
[1] "moment"
> kurtosis(min_total_cost)
[1] -0.5798372
attr(,"method")
[1] "excess"
```



To assess whether the minimum total costs follow a normal distribution, we also fitted a normal distribution to the data and conducted a chi-squared goodness-of-fit test to assess how well the normal distribution fits the observed data.

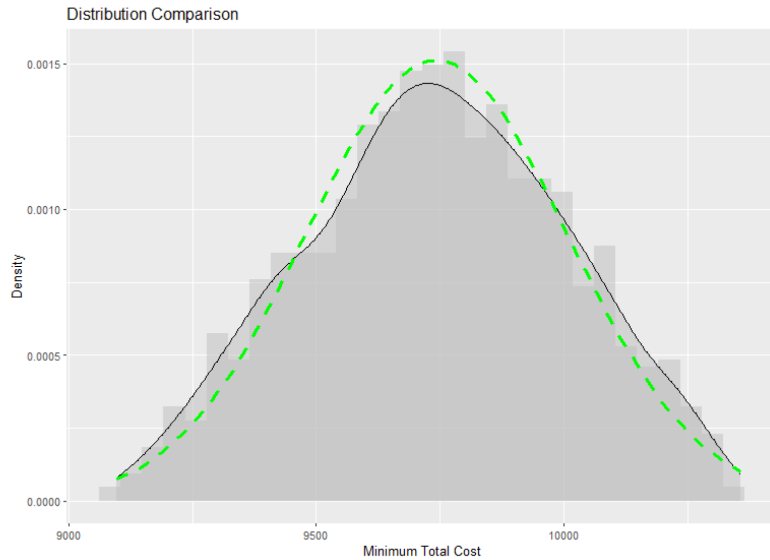
```
> # Create a data frame for the histogram plot
> df1 <- data.frame(min_total_cost)
>
> # Fit distributions
> library(MASS)
> fit_normal1 <- fitdistr(min_total_cost, "normal")
>
> # Create a histogram for 'min_total_cost'
> hist_min_total_cost <- hist(min_total_cost, breaks = 500, freq = FALSE)
> observed_counts <- hist_min_total_cost$counts
> bin_edges <- hist_min_total_cost$breaks
>
> # Calculate expected counts for normal distribution
> expected_normal <- diff(pnorm(bin_edges, mean = fit_normal1$estimate["mean"], sd = fit_normal1$estimate["sd"])) * length(min_total_cost)
>
> # Perform Chi-squared Goodness-of-fit test
> chi_square_normal <- chisq.test(observed_counts, p = expected_normal, rescale.p = TRUE)
```

```
> # Print the Chi-squared test result
> print(chi_square_normal)

        Chi-squared test for given probabilities

data:  observed_counts
X-squared = 627.52, df = 630, p-value = 0.5204
```

With a p-value of 0.5204, which is greater than the typical significance level of 0.05, there is no significant evidence to reject the null hypothesis that the data follows a normal distribution. Therefore, based on this test, the data “min_total_cost” appears to be reasonably well approximated by a normal distribution.

Finally, we generated a detailed plot using “ggplot2” to compare the empirical distribution to the fitted normal distribution visually.

```
> # Create the base plot
> library(ggplot2)
> p1 <- ggplot(df1, aes(x = min_total_cost)) +
+   geom_histogram(aes(y = ..density..), bins = 30, fill = "gray", alpha = 0.5) +
+   geom_density(aes(y = ..density..), fill = "gray", alpha = 0.5) +
+   ggtitle("Distribution Comparison") +
+   xlab("Minimum Total Cost") +
+   ylab("Density")
>
> # Add lines for PDF
> p1 <- p1 + stat_function(fun = dnorm, args = list(mean = fit_normal1$estimate["mean"], sd = fit_normal1$estimate["sd"]),
+                          color = "green", size = 1.2, linetype = "dashed")
>
> # Display the plot
> p1
```



Overall, the mean expected minimum total cost was estimated to be 9742.772, with a 95% confidence interval of [9726.393, 9759.151]. The normal distribution was found to be a suitable model for the data, as validated by both the Chi-squared goodness-of-fit test and the visual inspection of the distribution plot.

(ii) Estimate the expected order quantity by constructing a 95% confidence interval for it and determine the probability distribution that best fits its distribution. Verify the validity of your choice.

To estimate the expected order quantity, we began by calculating the mean and standard deviation of the observed optimal order quantities ("optimal_Q"). The mean optimal order quantity was 677.08. Using this mean, we calculated the standard error of the mean by dividing the standard deviation by the square root of the number of observations. This allowed me to construct a 95% confidence interval for the expected order quantity. The resulting confidence interval was [675.9416, 678.2184], indicating that we can be 95% confident that the true mean order quantity falls within this range.

```
> # (ii) Estimate expected order quantity and its 95% confidence interval
> mean_optimal_Q <- mean(optimal_Q)
> sd_optimal_Q <- sd(optimal_Q)
> se_optimal_Q <- sd_optimal_Q / sqrt(n_obs)
> ci_optimal_Q <- c(mean_optimal_Q - z_score * se_optimal_Q,
+                   mean_optimal_Q + z_score * se_optimal_Q)
>
> # Output the results
> cat("Estimated Expected Order Quantity:\n")
Estimated Expected Order Quantity:
> cat("Mean:", mean_optimal_Q, "\n")
Mean: 677.08
> cat("95% Confidence Interval:", ci_optimal_Q, "\n\n")
95% Confidence Interval: 675.9416 678.2184
```

Next, we plotted a histogram of the "optimal_Q" values to visualize their distribution. The histogram was complemented by a density plot to provide a smooth estimate of the distribution.

```
> # Histogram plot to visualize the distribution
> hist(optimal_Q, freq = FALSE, main = "Histogram of Expected Order Quantity")
```

We also calculated the skewness and kurtosis of the data to further understand its characteristics. The skewness was -0.0536, indicating a slight leftward skew, while the kurtosis was -0.5723, suggesting a distribution with lighter tails than a normal distribution.

```
> skewness(optimal_Q)
[1] -0.05361295
attr(,"method")
[1] "moment"
> kurtosis(optimal_Q)
[1] -0.5723345
attr(,"method")
[1] "excess"
```

To determine the probability distribution that best fits the observed data, we fitted a normal distribution to the "optimal_Q" values using the "fitdistr" function from the MASS package. We then performed a Chi-squared goodness-of-fit test to statistically assess the fit of the normal distribution.

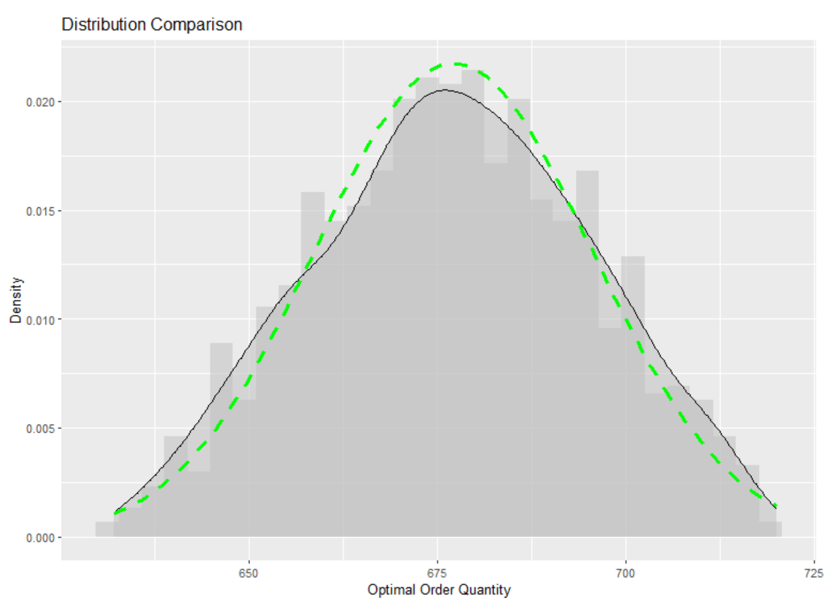
```
> # Create a data frame for the histogram plot
> df2 <- data.frame(optimal_Q)
>
> # Fit distributions
> library(MASS)
> fit_normal2 <- fitdistr(optimal_Q, "normal")
>
> # Create a histogram for 'optimal_Q'
> hist_optimal_Q <- hist(optimal_Q, breaks = 50, freq = FALSE)
> observed_counts <- hist_optimal_Q$counts
> bin_edges <- hist_optimal_Q$breaks
>
> # Calculate expected counts for normal distribution
> expected_normal <- diff(pnorm(bin_edges, mean = fit_normal2$estimate["mean"], sd = fit_normal2$estimate["sd"])) * length(optimal_Q)
>
> # Perform Chi-squared Goodness-of-fit test
> chi_square_normal <- chisq.test(observed_counts, p = expected_normal, rescale.p = TRUE)
```

```
> # Print the Chi-squared test result
> print(chi_square_normal)

        Chi-squared test for given probabilities

data:  observed_counts
X-squared = 49.212, df = 43, p-value = 0.2385
```

The test result indicated a Chi-squared statistic of 49.212 with 43 degrees of freedom, and a p-value of 0.2385. Since the p-value was greater than 0.05, we concluded that there was no significant difference between the observed and expected counts, thus supporting the normal distribution as a good fit for the data.

To visualize the fit, we created a plot that overlaid the fitted normal distribution on the histogram of the "optimal_Q" data. The plot included both the histogram and a dashed line representing the probability density function of the fitted normal distribution. This visual comparison, along with the statistical test results, confirmed the appropriateness of the normal distribution for modeling the expected order quantity.

```
> # Create the base plot
> p2 <- ggplot(df2, aes(x = optimal_Q)) +
+   geom_histogram(aes(y = ..density..), bins = 30, fill = "gray", alpha = 0.5) +
+   geom_density(aes(y = ..density..), fill = "gray", alpha = 0.5) +
+   ggtitle("Distribution Comparison") +
+   xlab("Optimal Order Quantity") +
+   ylab("Density")
>
> # Add lines for PDF
> p2 <- p2 + stat_function(fun = dnorm, args = list(mean = fit_normal2$estimate["mean"], sd = fit_normal2$estimate["sd"]),
+                          color = "green", size = 1.2, linetype = "dashed")
>
> # Display the plot
> p2
```



Overall, the mean expected order quantity was estimated to be 677.08, with a 95% confidence interval of [675.9416, 678.2184]. The normal distribution was found to be a suitable model for the data, as validated by both the Chi-squared goodness-of-fit test and the visual inspection of the distribution plot.

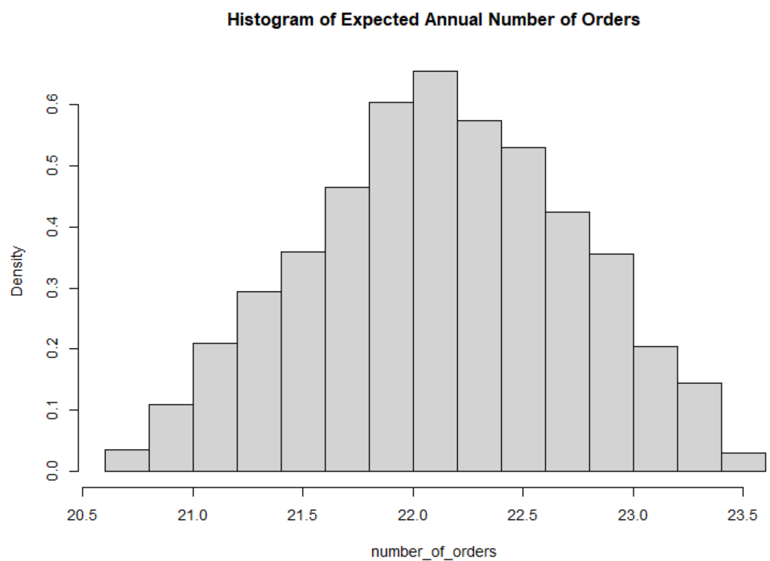
(iii) Estimate the expected annual number of orders by constructing a 95% confidence interval for it and determine the probability distribution that best fits its distribution. Verify the validity of your choice.

To estimate the expected annual number of orders, we first calculated the mean and standard deviation of the observed annual order quantities ("number_of_orders"). The mean annual order quantity was found to be 22.12634. Using this mean, we calculated the standard error of the mean by dividing the standard deviation by the square root of the number of observations. This allowed me to construct a 95%

confidence interval for the expected annual number of orders. The resulting confidence interval was [22.08914, 22.16355], indicating that we can be 95% confident that the true mean number of annual orders falls within this range.

```
> # (iii) Estimate expected annual number of orders and its 95% confidence interval
> mean_number_of_orders <- mean(number_of_orders)
> sd_number_of_orders <- sd(number_of_orders)
> se_number_of_orders <- sd_number_of_orders / sqrt(n_obs)
> ci_number_of_orders <- c(mean_number_of_orders - z_score * se_number_of_orders,
+                          mean_number_of_orders + z_score * se_number_of_orders)
>
> # Output the results
> cat("Estimated Expected Annual Number of Orders:\n")
Estimated Expected Annual Number of Orders:
> cat("Mean:", mean_number_of_orders, "\n")
Mean: 22.12634
> cat("95% Confidence Interval:", ci_number_of_orders, "\n")
95% Confidence Interval: 22.08914 22.16355
```

Next, we analyzed the distribution of the annual order quantities. By plotting a histogram of "number_of_orders" with a density overlay, I visualized the data's distribution. The histogram suggested the overall shape and spread of the data, while the density plot provided a smooth estimate of the distribution.



To further characterize the distribution, we calculated skewness and kurtosis. These metrics helped in understanding if the data deviated significantly from normality. The skewness was -0.0533, indicating a slight leftward skew, while the kurtosis was -0.5868, suggesting a distribution with lighter tails than a normal distribution.

```
> skewness(number_of_orders)
[1] -0.05329778
attr(,"method")
[1] "moment"
> kurtosis(number_of_orders)
[1] -0.5867757
attr(,"method")
[1] "excess"
```

We fit a normal distribution to the “number_of_orders” data using the “fitdistr” function from the MASS package.

```
> # Fit distributions
> library(MASS)
> fit_normal3 <- fitdistr(number_of_orders, "normal")
```

To verify the suitability of the normal distribution, we performed a Chi-squared goodness-of-fit test.

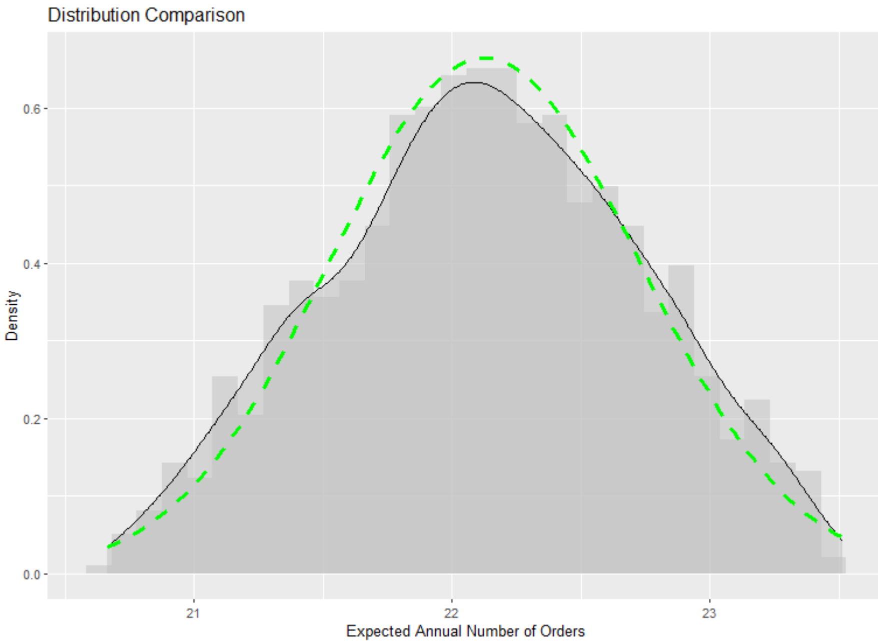
```
> # Create a histogram for 'number_of_orders'
> hist_number_of_orders <- hist(number_of_orders, breaks = 50, freq = FALSE)
> observed_counts <- hist_number_of_orders$counts
> bin_edges <- hist_number_of_orders$breaks
>
> # Calculate expected counts for normal distribution
> expected_normal <- diff(pnorm(bin_edges, mean = fit_normal3$estimate["mean"], sd = fit_normal3$estimate["sd"])) * length(number_of_orders)
>
> # Perform Chi-squared Goodness-of-fit test
> chi_square_normal <- chisq.test(observed_counts, p = expected_normal, rescale.p = TRUE)
```

```
> # Print the Chi-squared test result
> print(chi_square_normal)

        Chi-squared test for given probabilities

data:  observed_counts
X-squared = 60.877, df = 57, p-value = 0.3382
```

The test result indicated a Chi-squared statistic of 60.877 with 57 degrees of freedom, and a p-value of 0.3382. Since the p-value was greater than 0.05, we concluded that there was no significant difference between the observed and expected counts, thus supporting the normal distribution as a good fit for the data.

Finally, we visualized the fitted normal distribution against the histogram of the data. This comparison helped in assessing the fit qualitatively.

```
> # Create the base plot
> library(ggplot2)
> p3 <- ggplot(df1, aes(x = number_of_orders)) +
+   geom_histogram(aes(y = ..density..), bins = 30, fill = "gray", alpha = 0.5) +
+   geom_density(aes(y = ..density..), fill = "gray", alpha = 0.5) +
+   ggtitle("Distribution Comparison") +
+   xlab("Expected Annual Number of Orders") +
+   ylab("Density")
>
> # Add lines for PDF
> p3 <- p3 + stat_function(fun = dnorm, args = list(mean = fit_normal3$estimate["mean"], sd = fit_normal3$estimate["sd"]),
+                          color = "green", size = 1.2, linetype = "dashed")
>
> # Display the plot
> p3
```



Overall, the expected annual number of orders was estimated to be 22.12634, with a 95% confidence interval of [22.08914 22.16355]. The normal distribution was found to be a suitable model for the data, as validated by both the Chi-squared goodness-of-fit test and the visual inspection of the distribution plot.

3. Summary of Results

(i) Estimated Expected Minimum Total Cost:

- **Mean**: 9742.772
- **95% Confidence Interval**: [9726.393, 9759.151]
- **Best-fitting distribution**: Normal

(ii) Estimated Expected Order Quantity:

- **Mean**: 677.08
- **95% Confidence Interval**: [675.9416, 678.2184]
- **Best-fitting distribution**: Normal

(iii) Estimated Expected Annual Number of Orders:

- **Mean**: 22.12634
- **95% Confidence Interval**: [22.08914, 22.16355]
- **Best-fitting distribution**: Normal

```
> # Summarize the results
> cat("Estimated Expected Minimum Total Cost:\n")
Estimated Expected Minimum Total Cost:
> cat("Mean:", mean_min_total_cost, "\n")
Mean: 9742.772
> cat("95% Confidence Interval:", ci_min_total_cost, "\n")
95% Confidence Interval: 9726.393 9759.151
> cat("Best-fitting distribution for Estimated Expected Minimum Total Cost: Normal\n\n")
Best-fitting distribution for Estimated Expected Minimum Total Cost: Normal

>
> cat("Estimated Expected Order Quantity:\n")
Estimated Expected Order Quantity:
> cat("Mean:", mean_optimal_Q, "\n")
Mean: 677.08
> cat("95% Confidence Interval:", ci_optimal_Q, "\n")
95% Confidence Interval: 675.9416 678.2184
> cat("Best-fitting distribution for Estimated Expected Order Quantity: Normal\n\n")
Best-fitting distribution for Estimated Expected Order Quantity: Normal

>
> cat("Estimated Expected Annual Number of Orders:\n")
Estimated Expected Annual Number of Orders:
> cat("Mean:", mean_number_of_orders, "\n")
Mean: 22.12634
> cat("95% Confidence Interval:", ci_number_of_orders, "\n")
95% Confidence Interval: 22.08914 22.16355
> cat("Best-fitting distribution for Expected Annual Number of Orders: Normal\n\n")
Best-fitting distribution for Expected Annual Number of Orders: Normal
```

References

Bennett, T. (2024, January 10). Inventory levels. *Priceva*. https://priceva.com/blog/inventory-levels

Pienaar, A. (n.d.). How to calculate and maintain optimal inventory levels. *Cogsy*. https://cogsy.com/backordering/inventory-levels/