Project 1: Inventory Simulation and Statistical Analysis | Part 1

Safal Thapa

2024-10-08

Contents


```
library(ggplot2)
library(reshape2)
library(dplyr)
```

1 SIMULATION FOR FIRST CASE: SIMULATE FOR MONTHLY (4 WEEKS)

```
lambda <- (1 / 7)  # Average customer arrival rate (1 customer per week)
weeks <- 4       # Simulating for 1 month (4 weeks)
days <- 7 * weeks # Total days in the simulation
profit_per_scooter <- 200  # Profit for each scooter sold as given
cost_per_lost_sale <- 100   # Cost associated with losing a customer as given
overstock_cost_per_day <- 5
delivery_time <- 5

a_simulate_inventory <- function(order_when_out, fixed_delivery,
                                 customer_arrivals)
{
```

```
stock <- 1
total_customers <- 0
total_sales <- 0
lost_sales <- 0
overstock_days <- 0
total_profit <- 0
restock_day_counter <- 0

  for (day in 1:days)
{
    customer_arrival <- customer_arrivals[day]
    total_customers <- total_customers + customer_arrival

  if (customer_arrival > 0)
    {
    if (stock > 0)
      {
      scooters_sold <- min(stock, customer_arrival)
      stock <- stock - scooters_sold
      total_sales <- total_sales + scooters_sold
      total_profit <- total_profit + (profit_per_scooter * scooters_sold)
      } else
        {
          lost_sales <- lost_sales + customer_arrival
        }
    }

  if (order_when_out == 1 && stock == 0 && restock_day_counter == 0)
    {
      restock_day_counter <- delivery_time
    }

  if (restock_day_counter > 0)
    {
      restock_day_counter <- restock_day_counter - 1
    }

      # Restock if the counter reaches zero
  if (restock_day_counter == 0 && stock == 0)
    {
      stock <- stock + 1
    }

  if (fixed_delivery > 0 && (day %% fixed_delivery == 0))
    {
     stock <- stock + 1
    }

  if (stock > 1)
    {
      overstock_days <- overstock_days + (stock - 1)
      total_profit <- total_profit - overstock_cost_per_day * (stock - 1)
    }
```

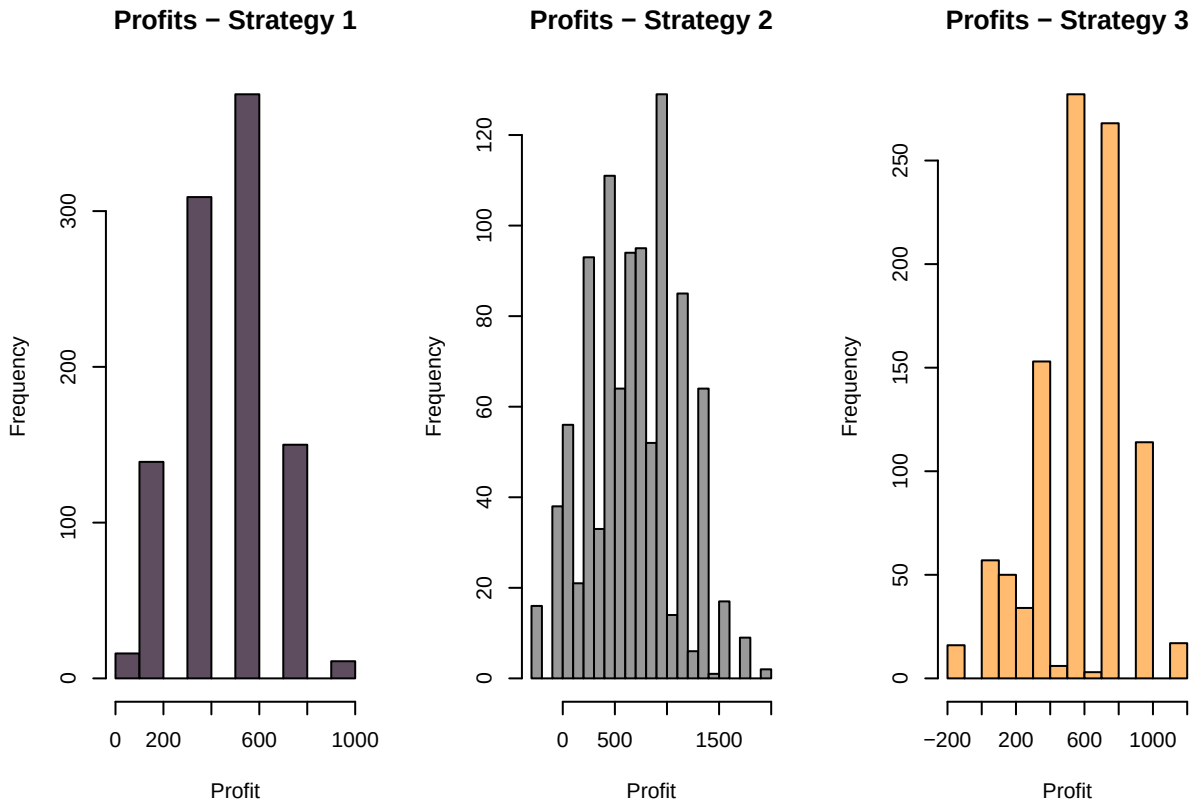
```
}
  fraction_served <- ifelse(total_customers > 0,
                            total_sales / total_customers, 0)
  if (total_customers > 0) {
        lost_sales <- (total_customers - total_sales) / total_customers
    } else {
        lost_sales <- 0
    }

  return(list(customers = total_customers, sales = total_sales,
              fraction_served = fraction_served,
              lost_sales_fraction = lost_sales,
              overstock_days = overstock_days,
              profit = total_profit))
}
```

```
# Define the number of simulations
n_simulations <- 1000

# Initializing empty lists to store results from each simulation
a_results_list <- list()

for (sim in 1:n_simulations) {

  # Generating customer arrivals for the simulation to make sure same
  # simulation get same customer arrival
  customer_arrivals <- rpois(days, lambda)
  # Simulate each strategy for this run
  a_strategy_1 <- a_simulate_inventory(order_when_out = 1, fixed_delivery = 0,
                                       customer_arrivals)
  a_strategy_2 <- a_simulate_inventory(order_when_out = 0, fixed_delivery = 7,
                                       customer_arrivals)
  a_strategy_3 <- a_simulate_inventory(order_when_out = 1, fixed_delivery = 10,
                                       customer_arrivals)

  a_simulation_results <- data.frame(
    Simulation = sim,
    Strategy = c("Order When Out of Stock", "Fixed Weekly Delivery",
                 "Hybrid Delivery (N=10)"),
    Customers = c(a_strategy_1$customers, a_strategy_2$customers,
                  a_strategy_3$customers),
    Sales = c(a_strategy_1$sales, a_strategy_2$sales, a_strategy_3$sales),
    Fraction_Served = c(a_strategy_1$fraction_served,
                        a_strategy_2$fraction_served,
                        a_strategy_3$fraction_served),
    Lost_Sales_Fraction = c(a_strategy_1$lost_sales_fraction,
                            a_strategy_2$lost_sales_fraction,
                            a_strategy_3$lost_sales_fraction),
    Overstock_Days = c(a_strategy_1$overstock_days, a_strategy_2$overstock_days,
                       a_strategy_3$overstock_days),
    Profit = c(a_strategy_1$profit, a_strategy_2$profit, a_strategy_3$profit)
  )
  a_results_list[[sim]] <- a_simulation_results
```

```
}

# Combine all results into one data frame
a_simulation_results <- do.call(rbind, a_results_list)
#print(a_simulation_results)
```

2 VISUALIZATION FOR FIRST SIMULATION: SIMULATE FOR MONTHLY (4 WEEKS)

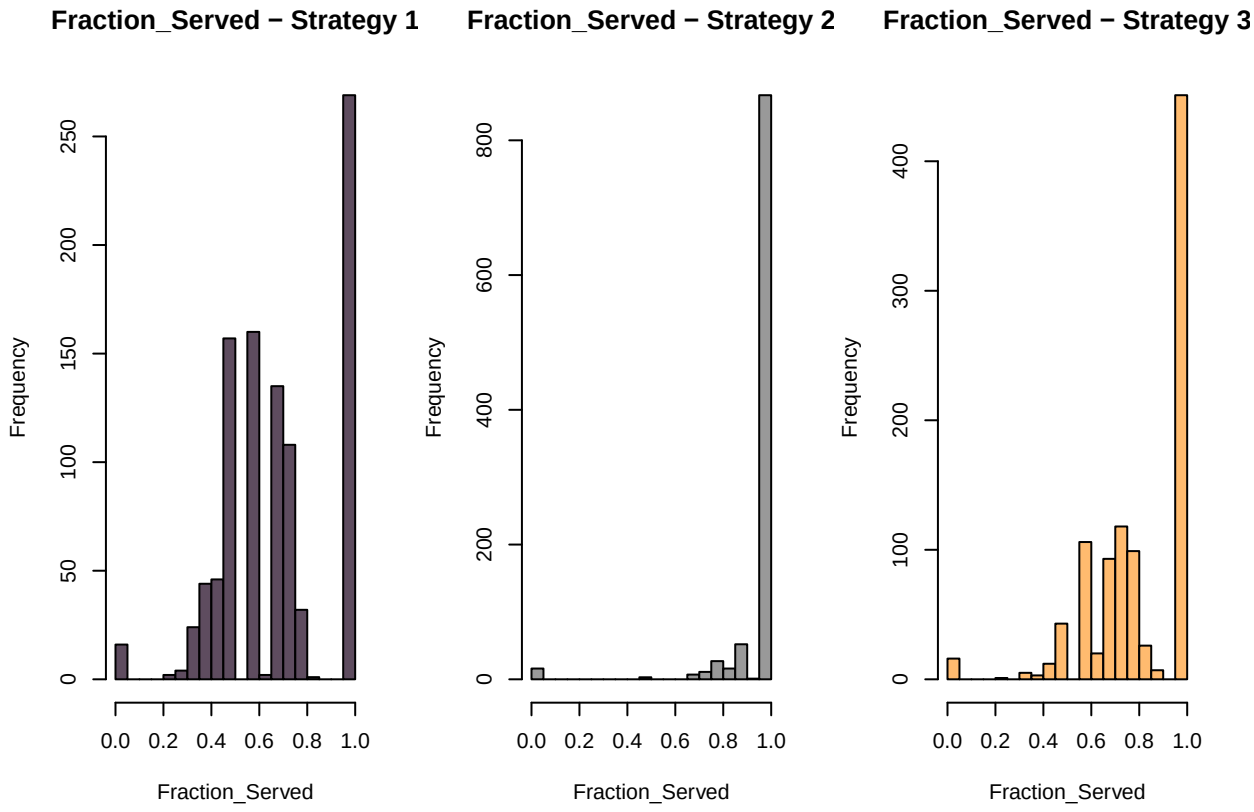
```
profits_strategy_1 <- a_simulation_results$Profit[
  a_simulation_results$Strategy == "Order When Out of Stock"]
profits_strategy_2 <- a_simulation_results$Profit[
  a_simulation_results$Strategy == "Fixed Weekly Delivery"]
profits_strategy_3 <- a_simulation_results$Profit[
  a_simulation_results$Strategy == "Hybrid Delivery (N=10)"]

par(mfrow = c(1, 3))
hist(profits_strategy_1,
     main = "Profits - Strategy 1",
     xlab = "Profit",
     ylab = "Frequency",
     col = "#5e4c5f",
     border = "black",
     breaks = 10)
hist(profits_strategy_2,
     main = "Profits - Strategy 2",
     xlab = "Profit",
     ylab = "Frequency",
     col = "#999999",
     border = "black",
     breaks = 20)
hist(profits_strategy_3,
     main = "Profits - Strategy 3",
     xlab = "Profit",
     ylab = "Frequency",
      col = "#ffbb6f",
     border = "black",
     breaks = 10)
```




```
par(mfrow = c(1, 1))
```

```
par(mfrow = c(1, 3))
fraction_served_strategy_1 <- a_simulation_results$Fraction_Served[
  a_simulation_results$Strategy == "Order When Out of Stock"]
fraction_served_strategy_2 <- a_simulation_results$Fraction_Served[
  a_simulation_results$Strategy == "Fixed Weekly Delivery"]
fraction_served_strategy_3 <- a_simulation_results$Fraction_Served[
  a_simulation_results$Strategy == "Hybrid Delivery (N=10)"]
hist(fraction_served_strategy_1,
     main = "Fraction_Served - Strategy 1",
     xlab = "Fraction_Served",
     ylab = "Frequency",
     col = "#5e4c5f",
     border = "black",
     breaks = 20)
hist(fraction_served_strategy_2,
     main = "Fraction_Served - Strategy 2",
     xlab = "Fraction_Served",
     ylab = "Frequency",
     col = "#999999",
     border = "black",
     breaks = 20)
hist(fraction_served_strategy_3,
     main = "Fraction_Served - Strategy 3",
```

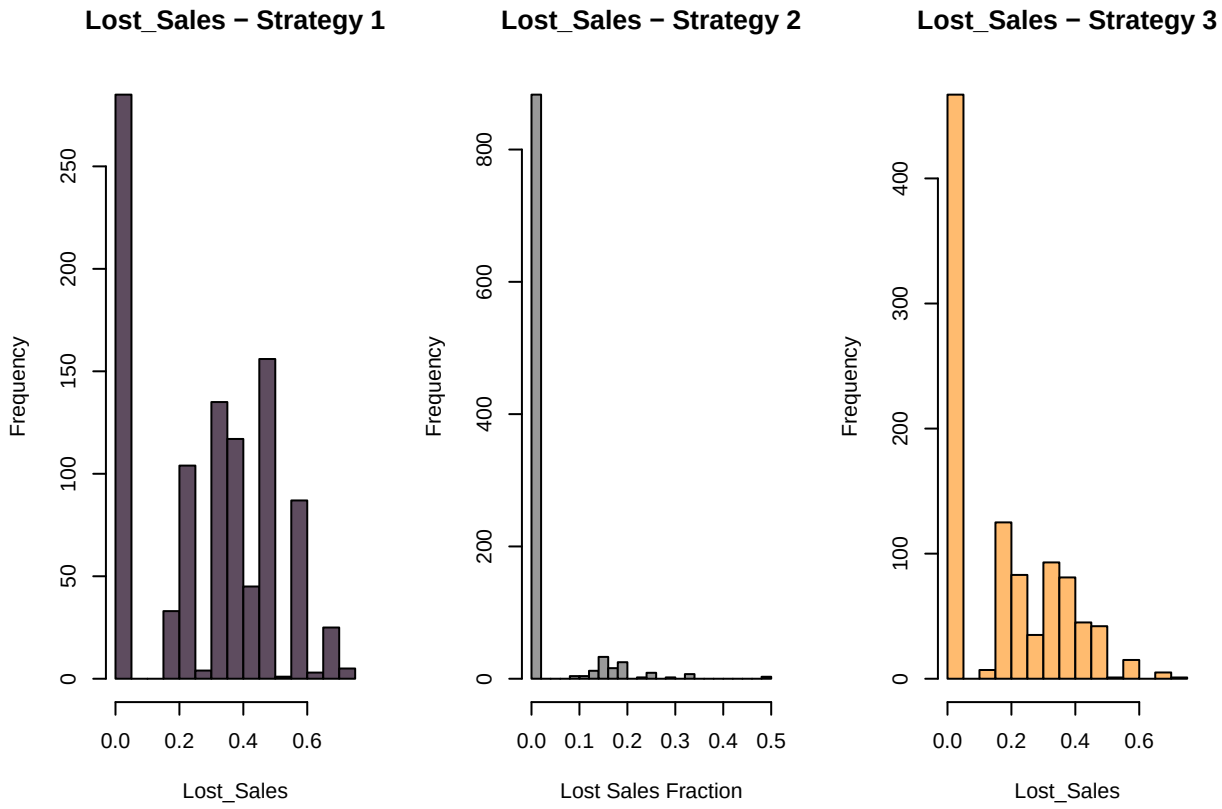
```
     xlab = "Fraction_Served",
     ylab = "Frequency",
     col = "#ffbb6f",
     border = "black",
     breaks = 20)
```




```
par(mfrow = c(1, 1))
```

```
par(mfrow = c(1, 3))
lost_sale_fraction_strategy_1 <- a_simulation_results$Lost_Sales_Fraction[
  a_simulation_results$Strategy == "Order When Out of Stock"]
lost_sale_fraction_strategy_2 <- a_simulation_results$Lost_Sales_Fraction[
  a_simulation_results$Strategy == "Fixed Weekly Delivery"]
lost_sale_fraction_strategy_3 <- a_simulation_results$Lost_Sales_Fraction[
  a_simulation_results$Strategy == "Hybrid Delivery (N=10)"]
hist(lost_sale_fraction_strategy_1,
     main = "Lost_Sales - Strategy 1",
     xlab = "Lost_Sales",
     ylab = "Frequency",
     col = "#5e4c5f",
     border = "black",
     breaks = 20)
hist(lost_sale_fraction_strategy_2,
     main = "Lost_Sales - Strategy 2",
     xlab = "Lost Sales Fraction",
```

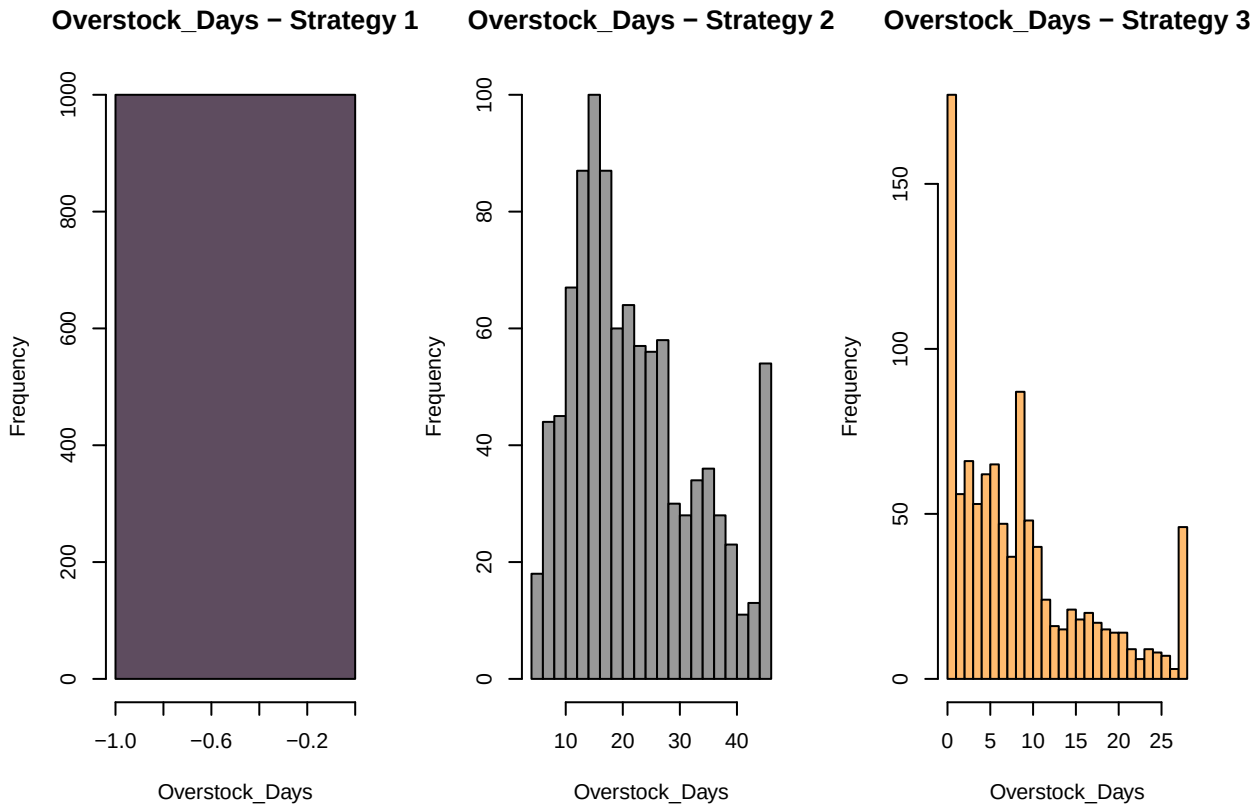
```
     ylab = "Frequency",
     col = "#999999",
     border = "black",
     breaks = 20)
hist(lost_sale_fraction_strategy_3,
     main = "Lost_Sales - Strategy 3",
     xlab = "Lost_Sales",
     ylab = "Frequency",
      col = "#ffbb6f",
     border = "black",
     breaks = 20)
```




```
par(mfrow = c(1, 1))
```

```
par(mfrow = c(1, 3))
over_stock_days_strategy_1 <- a_simulation_results$Overstock_Days[
  a_simulation_results$Strategy == "Order When Out of Stock"]
over_stock_days_strategy_2 <- a_simulation_results$Overstock_Days[
  a_simulation_results$Strategy == "Fixed Weekly Delivery"]
over_stock_days_strategy_3 <- a_simulation_results$Overstock_Days[
  a_simulation_results$Strategy == "Hybrid Delivery (N=10)"]

hist(over_stock_days_strategy_1,
     main = "Overstock_Days - Strategy 1",
     xlab = "Overstock_Days",
```

```
     ylab = "Frequency",
     col = "#5e4c5f",
     border = "black",
     breaks = 20)
hist(over_stock_days_strategy_2,
     main = "Overstock_Days - Strategy 2",
     xlab = "Overstock_Days",
     ylab = "Frequency",
     col = "#999999",
     border = "black",
     breaks = 20)
hist(over_stock_days_strategy_3,
     main = "Overstock_Days - Strategy 3",
     xlab = "Overstock_Days",
     ylab = "Frequency",
     col = "#ffbb6f",
     border = "black",
     breaks = 20)
```




```
par(mfrow = c(1, 1))
```

3 SIMULATION FOR SECOND CASE: SIMULATE FOR QUATERLY (13 WEEKS)

```
lambda <- (1 / 7)  # Average customer arrival rate (1 customer per week)
weeks <- 13       # Simulating for 3 month (13 weeks)
days <- 7 * weeks # Total days in the simulation
profit_per_scooter <- 200  # Profit for each scooter sold as given
cost_per_lost_sale <- 100   # Cost associated with losing a customer as given
overstock_cost_per_day <- 5
delivery_time <- 5

b_simulate_inventory <- function(order_when_out, fixed_delivery,
                                 customer_arrivals)
{
  stock <- 1
  total_customers <- 0
  total_sales <- 0
  lost_sales <- 0
  overstock_days <- 0
  total_profit <- 0
  restock_day_counter <- 0

    for (day in 1:days)
  {
     customer_arrival <- customer_arrivals[day]
    total_customers <- total_customers + customer_arrival

    if (customer_arrival > 0)
      {
      if (stock > 0)
      {
        scooters_sold <- min(stock, customer_arrival)
        stock <- stock - scooters_sold
        total_sales <- total_sales + scooters_sold
        total_profit <- total_profit + (profit_per_scooter * scooters_sold)
      } else
          {
            lost_sales <- lost_sales + customer_arrival
          }
      }

    if (order_when_out == 1 && stock == 0 && restock_day_counter == 0)
      {
      restock_day_counter <- delivery_time
      }

      if (restock_day_counter > 0)
        {
            restock_day_counter <- restock_day_counter - 1
        }

      if (restock_day_counter == 0 && stock == 0)
```

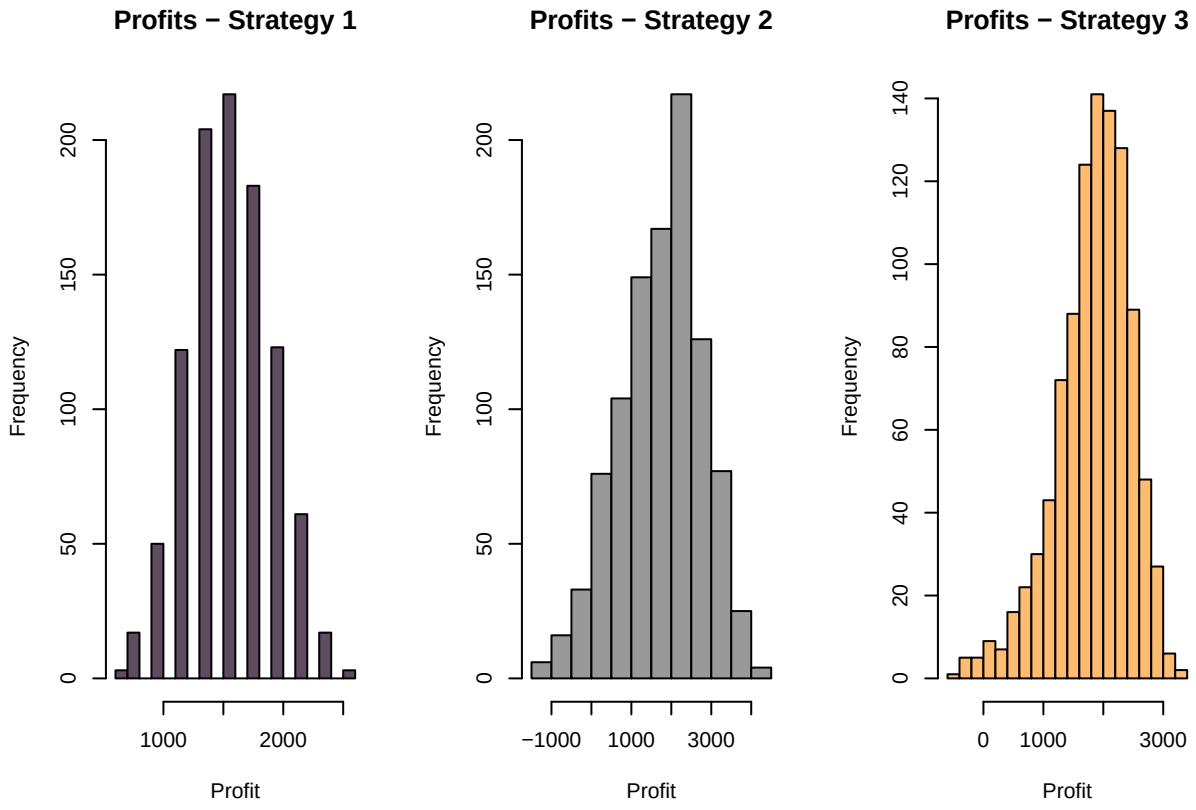
```
        {
          stock <- stock + 1
        }

      if (fixed_delivery > 0 && (day %% fixed_delivery == 0))
        {
         stock <- stock + 1
        }

      if (stock > 1)
        {
          overstock_days <- overstock_days + (stock - 1)
          total_profit <- total_profit - overstock_cost_per_day * (stock - 1)
        }
}
  fraction_served <- ifelse(total_customers > 0,
                            total_sales / total_customers, 0)
  if (total_customers > 0)
    {
        lost_sales <- (total_customers - total_sales) / total_customers
    } else
      {
        lost_sales <- 0
      }

  return(list(customers = total_customers, sales = total_sales,
              fraction_served = fraction_served,
              lost_sales_fraction = lost_sales,
              overstock_days = overstock_days,
              profit = total_profit))
}
```

```
n_simulations <- 1000
b_results_list <- list()

for (sim in 1:n_simulations) {
  customer_arrivals <- rpois(days, lambda)
  b_strategy_1 <- b_simulate_inventory(order_when_out = 1, fixed_delivery = 0,
                                       customer_arrivals)
  b_strategy_2 <- b_simulate_inventory(order_when_out = 0, fixed_delivery = 7,
                                       customer_arrivals)
  b_strategy_3 <- b_simulate_inventory(order_when_out = 1, fixed_delivery = 10,
                                       customer_arrivals)

  b_simulation_results <- data.frame(
    Simulation = sim,
    Strategy = c("Order When Out of Stock", "Fixed Weekly Delivery",
                 "Hybrid Delivery (N=10)"),
    Customers = c(b_strategy_1$customers, b_strategy_2$customers,
                  b_strategy_3$customers),
    Sales = c(b_strategy_1$sales, b_strategy_2$sales, b_strategy_3$sales),
    Fraction_Served = c(b_strategy_1$fraction_served,
                        b_strategy_2$fraction_served,
```

```
                        b_strategy_3$fraction_served),
    Lost_Sales_Fraction = c(b_strategy_1$lost_sales_fraction,
                            b_strategy_2$lost_sales_fraction,
                            b_strategy_3$lost_sales_fraction),
    Overstock_Days = c(b_strategy_1$overstock_days, b_strategy_2$overstock_days,
                       b_strategy_3$overstock_days),
    Profit = c(b_strategy_1$profit, b_strategy_2$profit, b_strategy_3$profit)
  )
  b_results_list[[sim]] <- b_simulation_results
}
# Combine all results into one data frame
b_simulation_results <- do.call(rbind, b_results_list)
#print(b_simulation_results)
```

4 VISUALIZATION FOR SECOND SIMULATION: SIMULATE FOR QUATERLY (13 WEEKS)

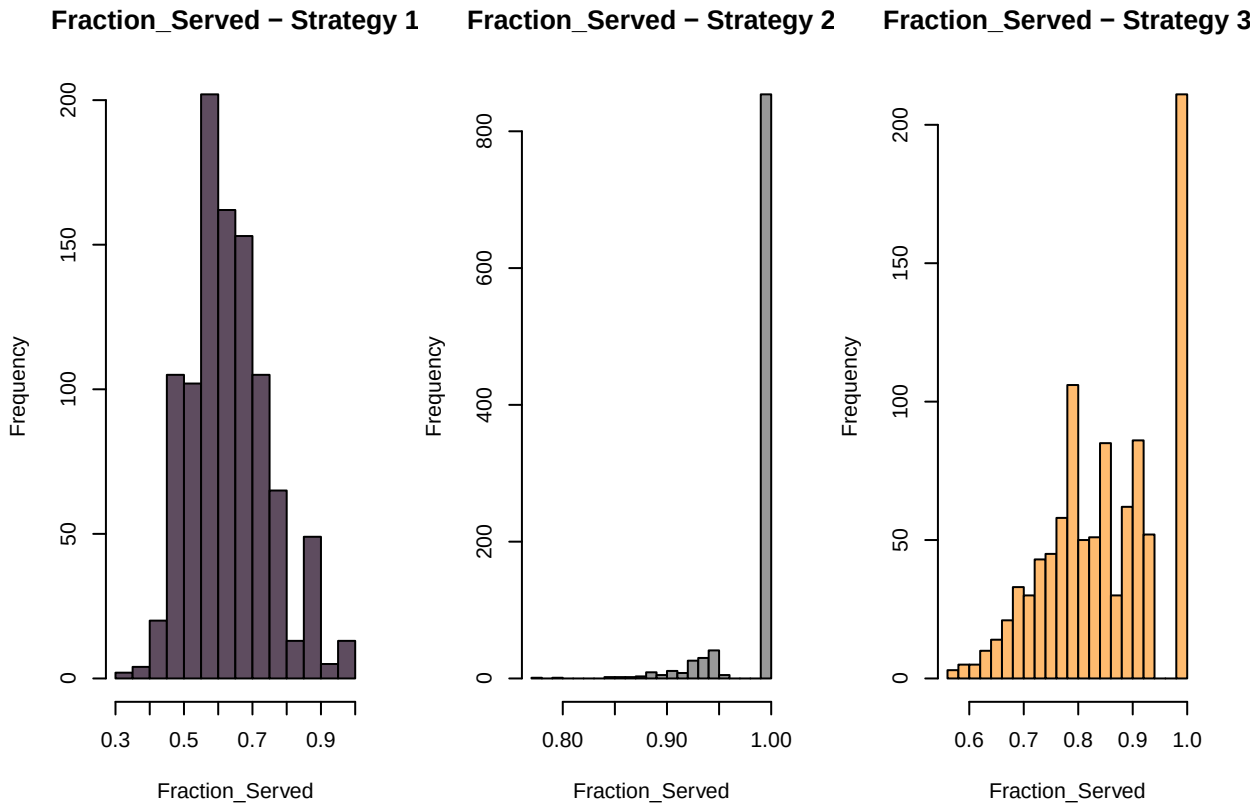
```
par(mfrow = c(1, 3))
b_profits_strategy_1 <- b_simulation_results$Profit[
  b_simulation_results$Strategy == "Order When Out of Stock"]
b_profits_strategy_2 <- b_simulation_results$Profit[
  b_simulation_results$Strategy == "Fixed Weekly Delivery"]
b_profits_strategy_3 <- b_simulation_results$Profit[
  b_simulation_results$Strategy == "Hybrid Delivery (N=10)"]

hist(b_profits_strategy_1,
     main = "Profits - Strategy 1",
     xlab = "Profit",
     ylab = "Frequency",
     col = "#5e4c5f",
     border = "black",
     breaks = 20)
hist(b_profits_strategy_2,
     main = "Profits - Strategy 2",
     xlab = "Profit",
     ylab = "Frequency",
     col = "#999999",
     border = "black",
     breaks = 20)
hist(b_profits_strategy_3,
     main = "Profits - Strategy 3",
     xlab = "Profit",
     ylab = "Frequency",
      col = "#ffbb6f",
     border = "black",
     breaks = 20)
```




```
par(mfrow = c(1, 1))
```

```
par(mfrow = c(1, 3))
b_fraction_served_strategy_1 <- b_simulation_results$Fraction_Served[
  b_simulation_results$Strategy == "Order When Out of Stock"]
b_fraction_served_strategy_2 <- b_simulation_results$Fraction_Served[
  b_simulation_results$Strategy == "Fixed Weekly Delivery"]
b_fraction_served_strategy_3 <- b_simulation_results$Fraction_Served[
  b_simulation_results$Strategy == "Hybrid Delivery (N=10)"]
hist(b_fraction_served_strategy_1,
     main = "Fraction_Served - Strategy 1",
     xlab = "Fraction_Served",
     ylab = "Frequency",
     col = "#5e4c5f",
     border = "black",
     breaks = 20)
hist(b_fraction_served_strategy_2,
     main = "Fraction_Served - Strategy 2",
     xlab = "Fraction_Served",
     ylab = "Frequency",
     col = "#999999",
     border = "black",
     breaks = 20)
hist(b_fraction_served_strategy_3,
     main = "Fraction_Served - Strategy 3",
```

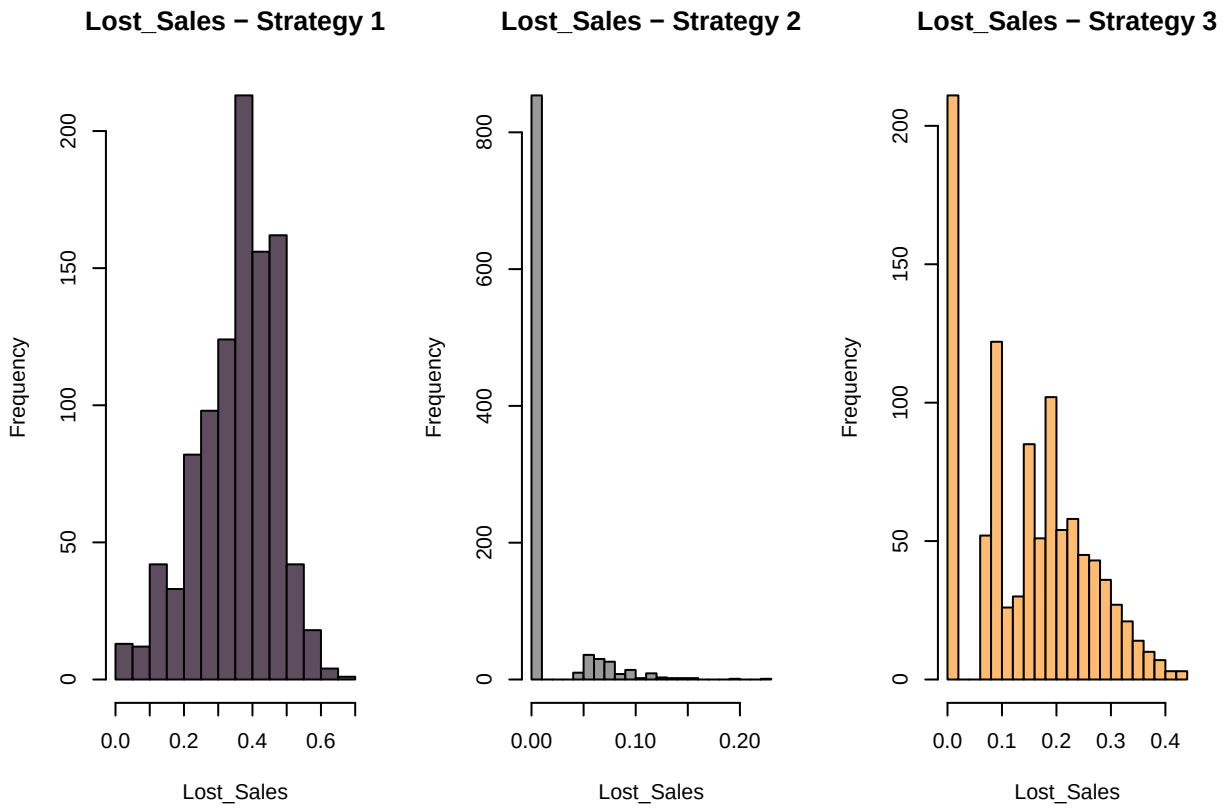
```
     xlab = "Fraction_Served",
     ylab = "Frequency",
     col = "#ffbb6f",
     border = "black",
     breaks = 20)
```




```
par(mfrow = c(1, 1))
```

```
par(mfrow = c(1, 3))
b_lost_sale_fraction_strategy_1 <- b_simulation_results$Lost_Sales_Fraction[
  b_simulation_results$Strategy == "Order When Out of Stock"]
b_lost_sale_fraction_strategy_2 <- b_simulation_results$Lost_Sales_Fraction[
  b_simulation_results$Strategy == "Fixed Weekly Delivery"]
b_lost_sale_fraction_strategy_3 <- b_simulation_results$Lost_Sales_Fraction[
  b_simulation_results$Strategy == "Hybrid Delivery (N=10)"]

hist(b_lost_sale_fraction_strategy_1,
     main = "Lost_Sales - Strategy 1",
     xlab = "Lost_Sales",
     ylab = "Frequency",
     col = "#5e4c5f",
     border = "black",
     breaks = 20)

hist(b_lost_sale_fraction_strategy_2,
```

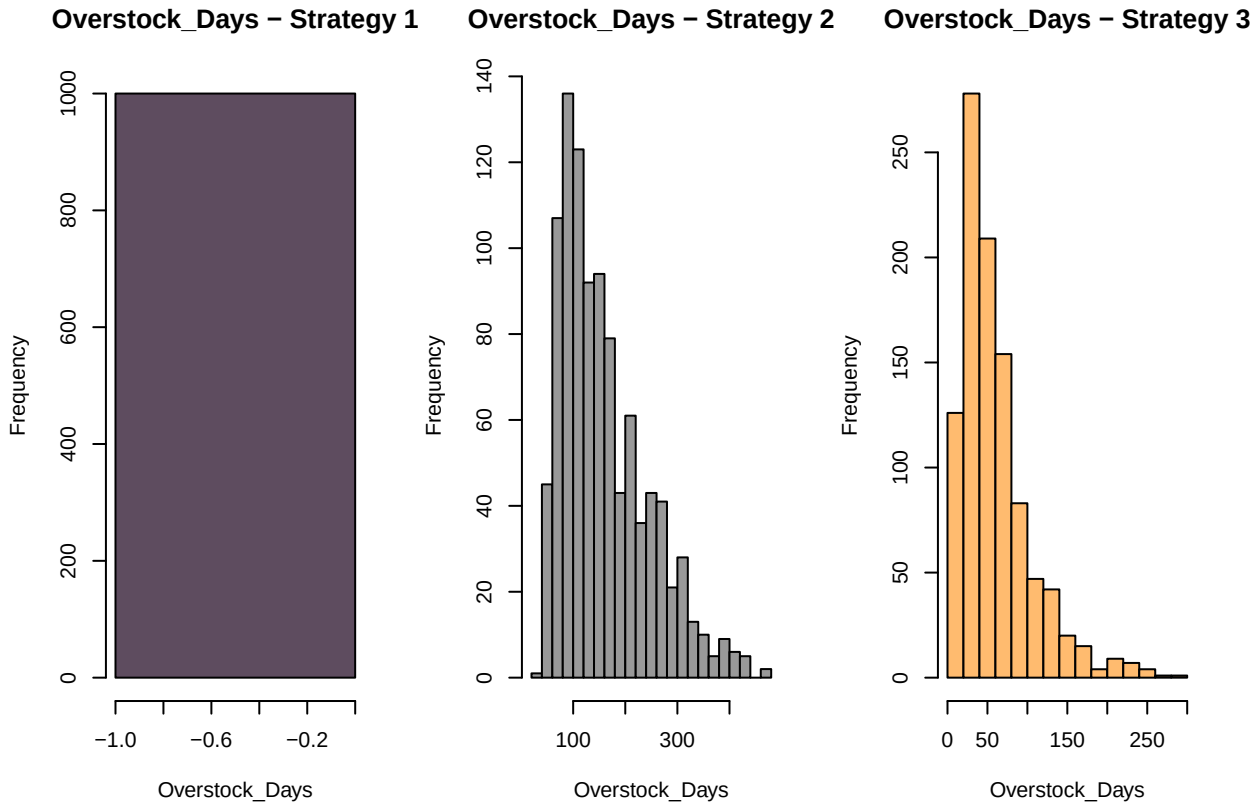
```
     main = "Lost_Sales - Strategy 2",
     xlab = "Lost_Sales",
     ylab = "Frequency",
     col = "#999999",
     border = "black",
     breaks = 20)
hist(b_lost_sale_fraction_strategy_3,
     main = "Lost_Sales - Strategy 3",
     xlab = "Lost_Sales",
     ylab = "Frequency",
      col = "#ffbb6f",
     border = "black",
     breaks = 20)
```




```
par(mfrow = c(1, 1))
```

```
par(mfrow = c(1, 3))
b_over_stock_days_strategy_1 <- b_simulation_results$Overstock_Days[
  b_simulation_results$Strategy == "Order When Out of Stock"]
b_over_stock_days_strategy_2 <- b_simulation_results$Overstock_Days[
  b_simulation_results$Strategy == "Fixed Weekly Delivery"]
b_over_stock_days_strategy_3 <- b_simulation_results$Overstock_Days[
  b_simulation_results$Strategy == "Hybrid Delivery (N=10)"]

hist(b_over_stock_days_strategy_1,
```

```
     main = "Overstock_Days - Strategy 1",
     xlab = "Overstock_Days",
     ylab = "Frequency",
     col = "#5e4c5f",
     border = "black",
     breaks = 20)
hist(b_over_stock_days_strategy_2,
     main = "Overstock_Days - Strategy 2",
     xlab = "Overstock_Days",
     ylab = "Frequency",
     col = "#999999",
     border = "black",
     breaks = 20)
hist(b_over_stock_days_strategy_3,
     main = "Overstock_Days - Strategy 3",
     xlab = "Overstock_Days",
     ylab = "Frequency",
     col = "#ffbb6f",
     border = "black",
     breaks = 20)
```




```
par(mfrow = c(1, 1))
```

5 SIMULATION FOR THIRD CASE: SIMULATE FOR YEARLY (52 WEEKS)

```
lambda <- (1 / 7)  # Average customer arrival rate (1 customer per week)
weeks <- 52       # Simulating for 3 month (13 weeks)
days <- (7 * weeks) # Total days in the simulation
profit_per_scooter <- 200  # Profit for each scooter sold as given
cost_per_lost_sale <- 100   # Cost associated with losing a customer as given
overstock_cost_per_day <- 5
delivery_time <- 5

c_simulate_inventory <- function(order_when_out, fixed_delivery,
                                 customer_arrivals)
{
  stock <- 1
  total_customers <- 0
  total_sales <- 0
  lost_sales <- 0
  overstock_days <- 0
  total_profit <- 0
  restock_day_counter <- 0

    for (day in 1:days)
  {
     customer_arrival <- customer_arrivals[day]
    total_customers <- total_customers + customer_arrival

    if (customer_arrival > 0)
      {
      if (stock > 0)
        {
        scooters_sold <- min(stock, customer_arrival)
        stock <- stock - scooters_sold
        total_sales <- total_sales + scooters_sold
        total_profit <- total_profit + (profit_per_scooter * scooters_sold)
        } else
          {
            lost_sales <- lost_sales + customer_arrival
          }
      }

    if (order_when_out == 1 && stock == 0 && restock_day_counter == 0)
      {
      restock_day_counter <- delivery_time
      }

    if (restock_day_counter > 0)
      {
      restock_day_counter <- restock_day_counter - 1
      }

        # Restock if the counter reaches zero
```

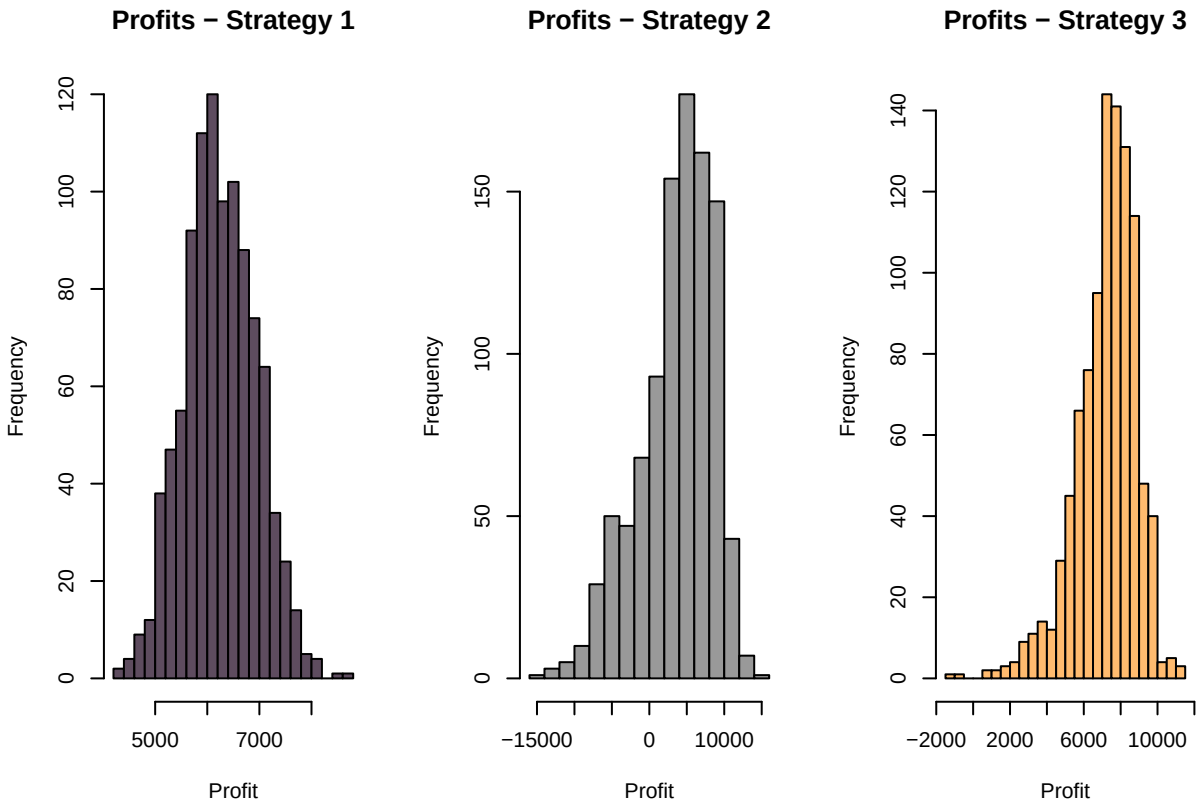
```
    if (restock_day_counter == 0 && stock == 0)
      {
      stock <- stock + 1
      }

    if (fixed_delivery > 0 && (day %% fixed_delivery == 0))
      {
      stock <- stock + 1
      }

    if (stock > 1)
      {
      overstock_days <- overstock_days + (stock - 1)
      total_profit <- total_profit - overstock_cost_per_day * (stock - 1)
      }

}
  fraction_served <- ifelse(total_customers > 0,
                            total_sales / total_customers, 0)
  if (total_customers > 0)
    {
        lost_sales <- (total_customers - total_sales) / total_customers
    } else
      {
        lost_sales <- 0
      }

  return(list(customers = total_customers, sales = total_sales,
              fraction_served = fraction_served,
              lost_sales_fraction = lost_sales,
              overstock_days = overstock_days,
              profit = total_profit))
}
```

```
n_simulations <- 1000
c_results_list <- list()
for (sim in 1:n_simulations) {
  customer_arrivals <- rpois(days, lambda)
  c_strategy_1 <- c_simulate_inventory(order_when_out = 1, fixed_delivery = 0,
                                       customer_arrivals)
  c_strategy_2 <- c_simulate_inventory(order_when_out = 0, fixed_delivery = 7,
                                       customer_arrivals)
  c_strategy_3 <- c_simulate_inventory(order_when_out = 1, fixed_delivery = 10,
                                       customer_arrivals)

  c_simulation_results <- data.frame(
    Simulation = sim,
    Strategy = c("Order When Out of Stock", "Fixed Weekly Delivery",
                 "Hybrid Delivery (N=10)"),
    Customers = c(c_strategy_1$customers, c_strategy_2$customers,
                  c_strategy_3$customers),
    Sales = c(c_strategy_1$sales, c_strategy_2$sales, c_strategy_3$sales),
    Fraction_Served = c(c_strategy_1$fraction_served,
```

```
                        c_strategy_2$fraction_served,
                        c_strategy_3$fraction_served),
    Lost_Sales_Fraction = c(c_strategy_1$lost_sales, c_strategy_2$lost_sales,
                            c_strategy_3$lost_sales),
    Overstock_Days = c(c_strategy_1$overstock_days, c_strategy_2$overstock_days,
                       c_strategy_3$overstock_days),
    Profit = c(c_strategy_1$profit, c_strategy_2$profit, c_strategy_3$profit)
  )
  c_results_list[[sim]] <- c_simulation_results
}
c_simulation_results <- do.call(rbind, c_results_list)
#print(c_simulation_results)
```

6 VISUALIZATION FOR THIRD SIMULATION: SIMULATE FOR YEARLY (52 WEEKS)

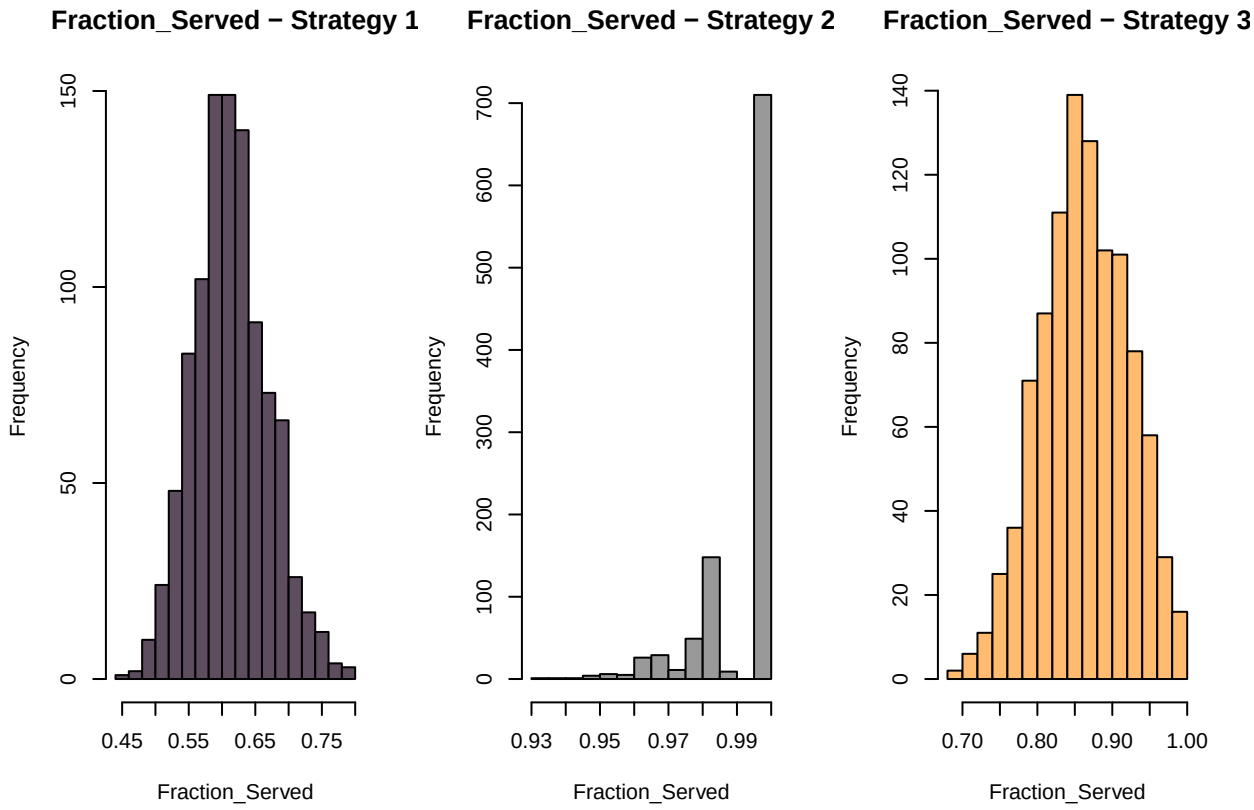
```
par(mfrow = c(1, 3))
c_profits_strategy_1 <- c_simulation_results$Profit[
  c_simulation_results$Strategy == "Order When Out of Stock"]
c_profits_strategy_2 <- c_simulation_results$Profit[
  c_simulation_results$Strategy == "Fixed Weekly Delivery"]
c_profits_strategy_3 <- c_simulation_results$Profit[
  c_simulation_results$Strategy == "Hybrid Delivery (N=10)"]

hist(c_profits_strategy_1,
     main = "Profits - Strategy 1",
     xlab = "Profit",
     ylab = "Frequency",
     col = "#5e4c5f",
     border = "black",
     breaks = 20)
hist(c_profits_strategy_2,
     main = "Profits - Strategy 2",
     xlab = "Profit",
     ylab = "Frequency",
     col = "#999999",
     border = "black",
     breaks = 20)
hist(c_profits_strategy_3,
     main = "Profits - Strategy 3",
     xlab = "Profit",
     ylab = "Frequency",
      col = "#ffbb6f",
     border = "black",
     breaks = 20)
```




```
par(mfrow = c(1, 1))
```

```
par(mfrow = c(1, 3))
c_fraction_served_strategy_1 <- c_simulation_results$Fraction_Served[
  c_simulation_results$Strategy == "Order When Out of Stock"]
c_fraction_served_strategy_2 <- c_simulation_results$Fraction_Served[
  c_simulation_results$Strategy == "Fixed Weekly Delivery"]
c_fraction_served_strategy_3 <- c_simulation_results$Fraction_Served[
  c_simulation_results$Strategy == "Hybrid Delivery (N=10)"]
hist(c_fraction_served_strategy_1,
     main = "Fraction_Served - Strategy 1",
     xlab = "Fraction_Served",
     ylab = "Frequency",
     col = "#5e4c5f",
     border = "black",
     breaks = 20)
hist(c_fraction_served_strategy_2,
     main = "Fraction_Served - Strategy 2",
     xlab = "Fraction_Served",
     ylab = "Frequency",
     col = "#999999",
     border = "black",
     breaks = 20)
hist(c_fraction_served_strategy_3,
     main = "Fraction_Served - Strategy 3",
```

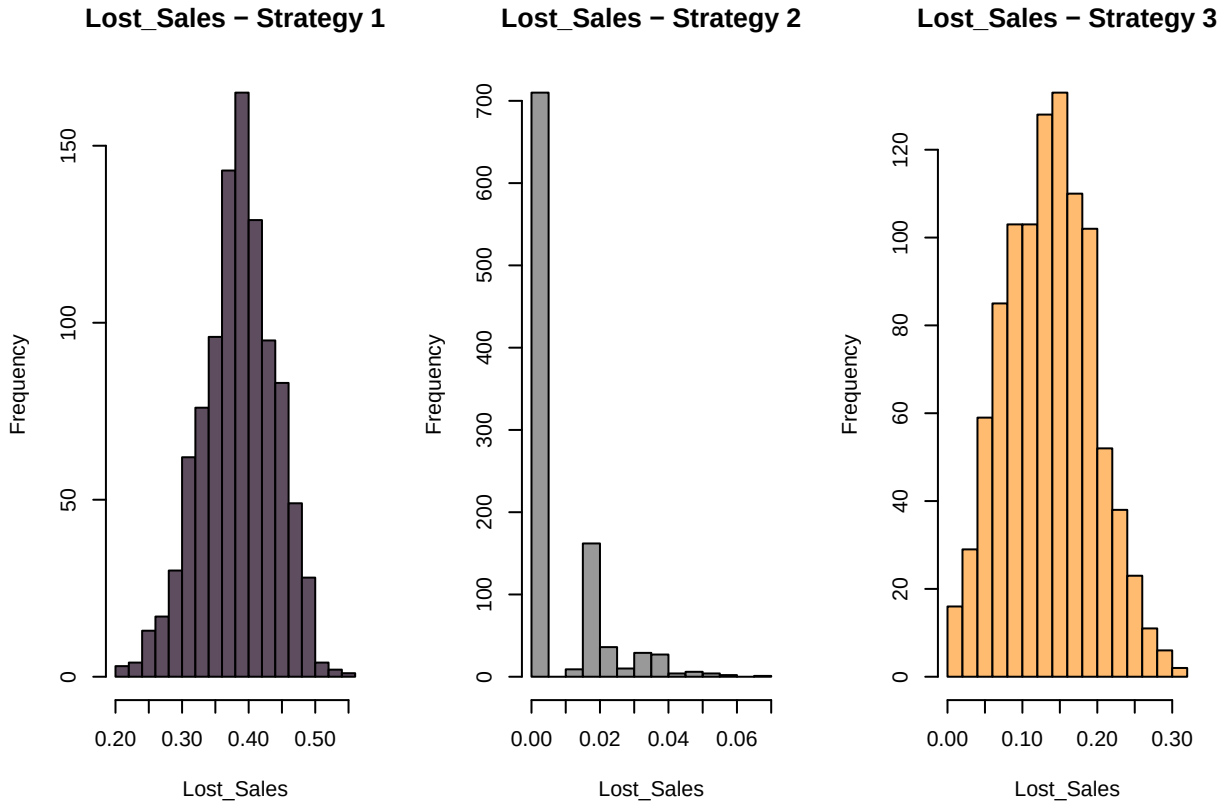
```
     xlab = "Fraction_Served",
     ylab = "Frequency",
     col = "#ffbb6f",
     border = "black",
     breaks = 20)
```




```
par(mfrow = c(1, 1))
```

```
par(mfrow = c(1, 3))
c_lost_sale_fraction_strategy_1 <- c_simulation_results$Lost_Sales_Fraction[
  c_simulation_results$Strategy == "Order When Out of Stock"]
c_lost_sale_fraction_strategy_2 <- c_simulation_results$Lost_Sales_Fraction[
  c_simulation_results$Strategy == "Fixed Weekly Delivery"]
c_lost_sale_fraction_strategy_3 <- c_simulation_results$Lost_Sales_Fraction[
  c_simulation_results$Strategy == "Hybrid Delivery (N=10)"]
hist(c_lost_sale_fraction_strategy_1,
     main = "Lost_Sales - Strategy 1",
     xlab = "Lost_Sales",
     ylab = "Frequency",
     col = "#5e4c5f",
     border = "black",
     breaks = 20)
hist(c_lost_sale_fraction_strategy_2,
     main = "Lost_Sales - Strategy 2",
     xlab = "Lost_Sales",
```

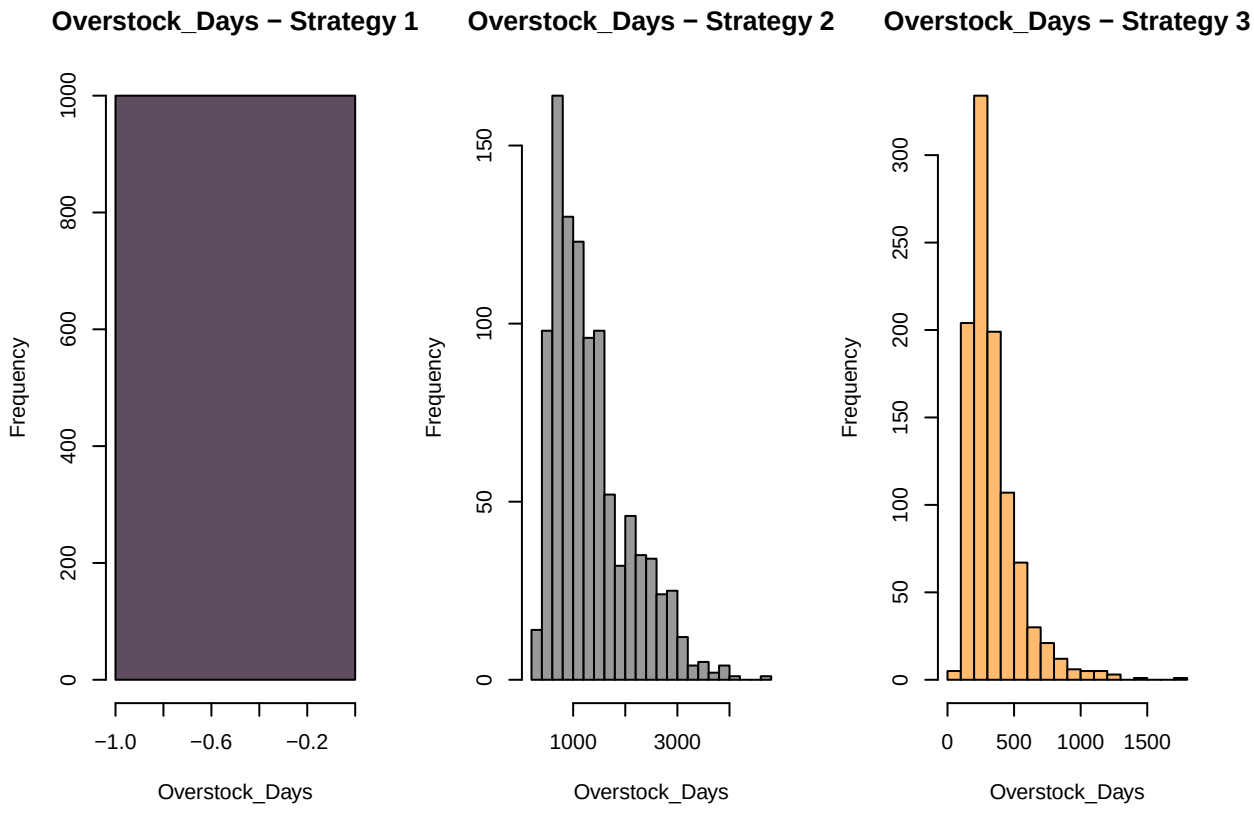
```
     ylab = "Frequency",
     col = "#999999",
     border = "black",
     breaks = 20)
hist(c_lost_sale_fraction_strategy_3,
     main = "Lost_Sales - Strategy 3",
     xlab = "Lost_Sales",
     ylab = "Frequency",
      col = "#ffbb6f",
     border = "black",
     breaks = 20)
```




```
par(mfrow = c(1, 1))
```

```
par(mfrow = c(1, 3))
c_over_stock_days_strategy_1 <- c_simulation_results$Overstock_Days[
  c_simulation_results$Strategy == "Order When Out of Stock"]

c_over_stock_days_strategy_2 <- c_simulation_results$Overstock_Days[
  c_simulation_results$Strategy == "Fixed Weekly Delivery"]

c_over_stock_days_strategy_3 <- c_simulation_results$Overstock_Days[
  c_simulation_results$Strategy == "Hybrid Delivery (N=10)"]
hist(c_over_stock_days_strategy_1,
     main = "Overstock_Days - Strategy 1",
```

```
     xlab = "Overstock_Days",
     ylab = "Frequency",
     col = "#5e4c5f",
     border = "black",
     breaks = 20)
hist(c_over_stock_days_strategy_2,
     main = "Overstock_Days - Strategy 2",
     xlab = "Overstock_Days",
     ylab = "Frequency",
     col = "#999999",
     border = "black",
     breaks = 20)
hist(c_over_stock_days_strategy_3,
     main = "Overstock_Days - Strategy 3",
     xlab = "Overstock_Days",
     ylab = "Frequency",
     col = "#ffbb6f",
     border = "black",
     breaks = 20)
```




```
par(mfrow = c(1, 1))
```