Optimizing Supply Chain Logistics

Felipe O. Cerezer

2024-09-15

Step 1. Required Libraries

We will use the following R packages: - `lpSolve` and `ompr` for solving optimization problems - `magrittr` for its pipe operator - `ggplot2` for data visualization

```
## Install required packages if not already installed
#install.packages(c("lpSolve", "ompr", "ompr.roi", "ROI.plugin.glpk", "ggplot2", "magrittr"))

## Load libraries
library(lpSolve)
library(ompr)
library(ompr.roi)
library(ROI.plugin.glpk)
library(ggplot2)
library(reshape2)
library(magrittr)
```

Step 2. Define the Optimization Problem

First, I will set up the cost matrix, supply, and demand for the optimization problem. This example involves optimizing transportation costs between three warehouses and four stores.

```
set.seed(42) # For reproducibility

# Define the cost matrix, supply, and demand
costs <- matrix(c(2, 4, 5, 3,
                  3, 1, 6, 4,
                  4, 2, 7, 5),
                nrow = 3, byrow = TRUE)  # Costs of transportation from warehouses to stores

supply <- c(20, 30, 25)                 # Supply available at each warehouse
demand <- c(10, 15, 25, 25)             # Demand required at each store
```

Explanation: Here, I define the parameters for our optimization problem: the costs matrix shows the transportation costs from each warehouse to each store, supply indicates how much each warehouse can provide, and demand represents the needs of each store. This setup will be used in both optimization methods.

Step 3: Solve the Problem Using lpSolve

Next, I will use the lpSolve package to minimize the total transportation cost with a linear programming model.

```
# Solve using lpSolve's transportation function
solution_lp <- lp.transport(costs, direction = "min",
                            row.signs = rep("<=", 3), row.rhs = supply,
                            col.signs = rep(">=", 4), col.rhs = demand)

# Print the optimal solution
print("Optimal transportation plan using lpSolve:")
```

```
## [1] "Optimal transportation plan using lpSolve:"
```

```
print(solution_lp$solution)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    0    0   20    0
## [2,]    0    0    5   25
## [3,]   10   15    0    0
```

Explanation: Here, I solve the optimization problem using the lpSolve package, which provides a straightforward way to handle linear programming problems. The function lp.transport is used to find the optimal transportation plan that minimizes the total cost. We print the result to see the distribution of resources.

Step 4: Solve the Problem Using ompr

I will use the ompr package for a more scalable solution. This step includes setting up the optimization model and solving it with the GLPK solver.

```
# Create the model using ompr
model <- MIPModel() %>%
  add_variable(x[i, j], i = 1:3, j = 1:4, type = "continuous", lb = 0) %>%
  set_objective(sum_expr(costs[i, j] * x[i, j], i = 1:3, j = 1:4), "min") %>%
  add_constraint(sum_expr(x[i, j], j = 1:4) <= supply[i], i = 1:3) %>%
  add_constraint(sum_expr(x[i, j], i = 1:3) >= demand[j], j = 1:4)

# Solve the model using GLPK solver
result_ompr <- solve_model(model, with_ROI(solver = "glpk"))

# Extract the solution values
solution_ompr <- get_solution(result_ompr, x[i, j])

# Print the optimal transportation plan from ompr
print("Optimal transportation plan using ompr:")
```

```
## [1] "Optimal transportation plan using ompr:"
```

```
print(solution_ompr)
```

```
##    variable i j value
## 1         x 1 1    10
## 5         x 2 1     0
## 9         x 3 1     0
## 2         x 1 2     0
## 6         x 2 2    15
## 10        x 3 2     0
## 3         x 1 3    10
## 7         x 2 3    15
## 11        x 3 3     0
## 4         x 1 4     0
## 8         x 2 4     0
## 12        x 3 4    25
```

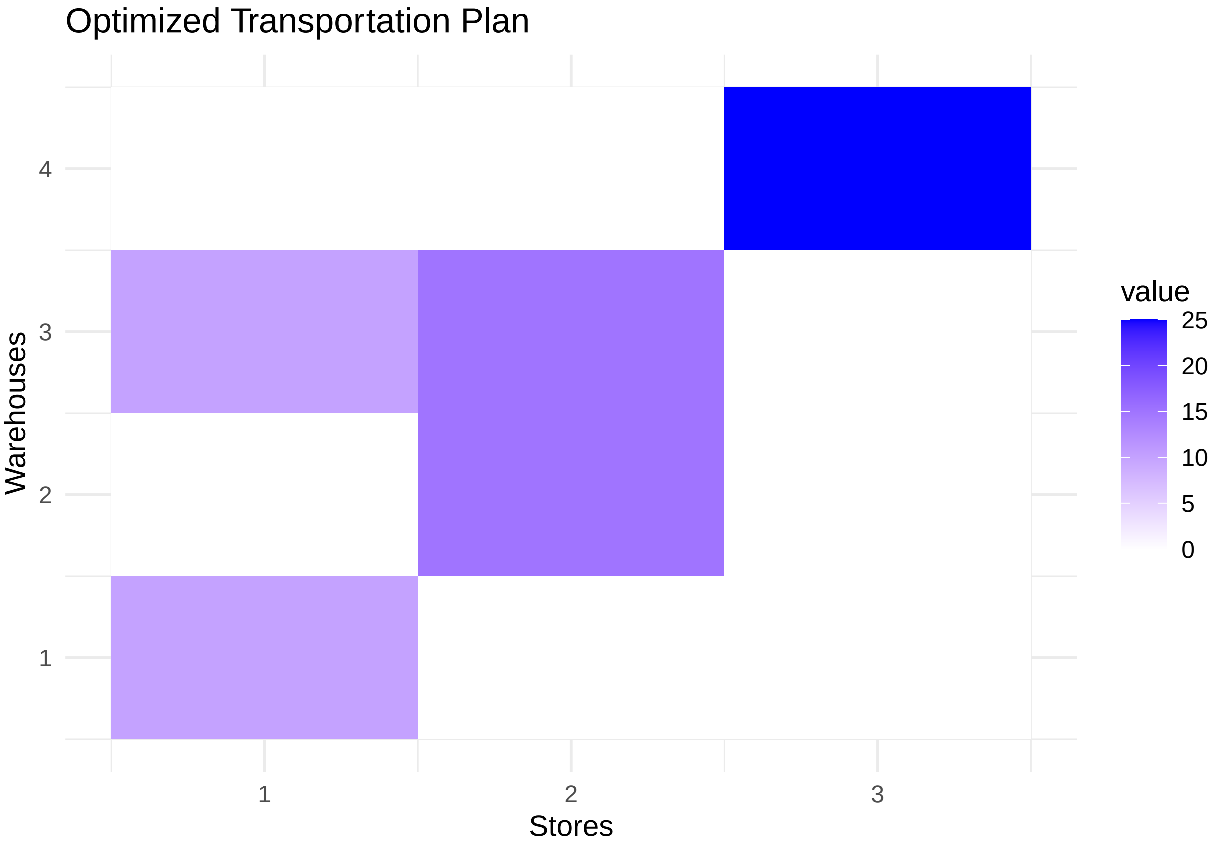
Explanation: In this step, I use the ompr package to create a more flexible and scalable optimization model. I define the decision variables, set the objective to minimize costs, and add constraints for supply and demand. The model is solved using the GLPK solver, and I extract and print the solution.

Step 5: Visualize the Results

Now, I will create a heatmap to visualize how much is transported from each warehouse to each store. This helps in understanding the distribution of resources.

```
# Convert the solution into a matrix format suitable for visualization
solution_matrix <- acast(solution_ompr, i ~ j, value.var = "value")

# Plot the optimized transportation plan
ggplot(melt(solution_matrix), aes(Var1, Var2, fill = value)) +
  geom_tile() +
  labs(x = "Stores", y = "Warehouses", title = "Optimized Transportation Plan") +
  scale_fill_gradient(low = "white", high = "blue") +
  theme_minimal()
```



Explanation: I convert the solution from the ompr model into a matrix format suitable for visualization. Then, I use ggplot2 to create a heatmap, which visually represents the amount of goods transported from each warehouse to each store. The resulting heatmap visually displays how resources are allocated from warehouses to stores, making it easy to see which warehouses supply which stores and the quantity transported. Darker shades represent higher quantities, allowing quick assessment of the distribution and efficiency of the transportation plan.