

Final Project - Quantitative Management Modeling

Laura Sasaki, Dibakar Bhowal, Gopinadh Ainala

2024-12-06

Purpose

Define, formulate, and solve mathematical optimization model.

Problem

The mathematical model to be optimized is focused on creating groups for a class of students. The parameters given were that the class has 15 students, and each group should have exactly 3 students. The idea is that each group would maximize their combined skills.

Variables

$x_{ij} = 1$ if the student is assigned to group j

$x_{ij} = 0$ if the student is not assigned to group j

Where i represents each available student and j represents the different groups formed.

$i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15$

$j = 1, 2, 3, 4, 5$

Constraints

Student assignment constraint

$$x_{i1} + x_{i2} + x_{i3} + x_{i4} + x_{i5}$$

Where $i = 1 : 15$

$$\sum_{j=1}^5 x_{ij} = 1, \quad \text{for } i = 1, 2, \dots, 15$$

Group size constraint

$$x_{1j} + x_{2j} + x_{3j} + x_{4j} + x_{5j} + x_{6j} + x_{7j} + x_{8j} + x_{9j} + x_{10j} + x_{11j} + x_{12j} + x_{13j} + x_{14j} + x_{15j}$$

Where $j = 1 : 5$

$$\sum_{i=1}^{15} x_{ij} = 3, \quad \text{for } j = 1, 2, \dots, 5$$

Binary decision variables constraint

$$x_{ij} \in \{0, 1\}$$

Where $i = 1 : 15$ and $j = 1 : 5$

Objective function

$$\text{Maximize } Z = \sum_{i=1}^{15} \sum_{j=1}^5 \text{SuccessScore}_i \cdot x_{ij}$$

Where, SuccessScore_i is the weighted average based on the factors (CommunicationSkills, ProblemSolving, and PresentationSkills) for student i .

These problem would take 3 main steps, generating random data, the mathematical formulation, and solving the problem set.

As always, any necessary libraries have to be called and, if necessary, installed prior to solving the problem. In this case the *lpSolve* and *ggplot2* library are the ones to be used.

```
library(lpSolve)
library(ggplot2)
```

Step 1

Step 1, as mentioned previously, consists of generating the data for the set problem. In this case setting data for 15 students, with a random score between 5.0 and 10.0 for all factors. The *set.seed* function was applied prior to ensure that the result can be duplicated, and then the *Students* data frame will be created.

Subsequently, the weighted success scores are calculated. The weight assigned to each factor were determined to be .4,.4,.2 as Communication skills and Problem solving are generally more frequently required than Presentation skills. The success score sums each of the factors with their weight and assigns it to the student.

```
# Step 1: Generate Random Student Data
set.seed(123)
n_students <- 15
n_groups <- 5
students <- data.frame(
  ID = 1:n_students,
  CommunicationSkills = runif(n_students, 5, 10),
  ProblemSolving = runif(n_students, 5, 10),
  PresentationSkills = runif(n_students, 5, 10)
)

# Step 2: Calculate Success Scores based on predefined weights
weights <- c(0.4, 0.4, 0.2)
students$SuccessScore <- with(
  students,
  CommunicationSkills * weights[1] +
  ProblemSolving * weights[2] +
  PresentationSkills * weights[3]
)
print("Generated Student Data:")

## [1] "Generated Student Data:

print(students)

## #> #>   ID CommunicationSkills ProblemSolving PresentationSkills SuccessScore
## #> 1   1          6.437888     9.499125      9.815121     8.337829
## #> 2   2          8.941526     6.230439      9.511495     7.971085
## #> 3   3          7.044885     5.210298      8.453526     6.592778
## #> 4   4          9.415087     6.639604      8.977337     8.217344
## #> 5   5          9.702336     9.772518      5.123068     8.814556
## #> 6   6          5.227782     9.447697      7.388980     7.347988
## #> 7   7          7.640527     8.464017      8.792298     8.200277
## #> 8   8          9.462095     8.202534      6.082040     8.282260
## #> 9   9          7.757175     9.971349      6.590905     8.409591
## #> 10 10         7.283074     8.278529      6.158129     7.456267
```

```

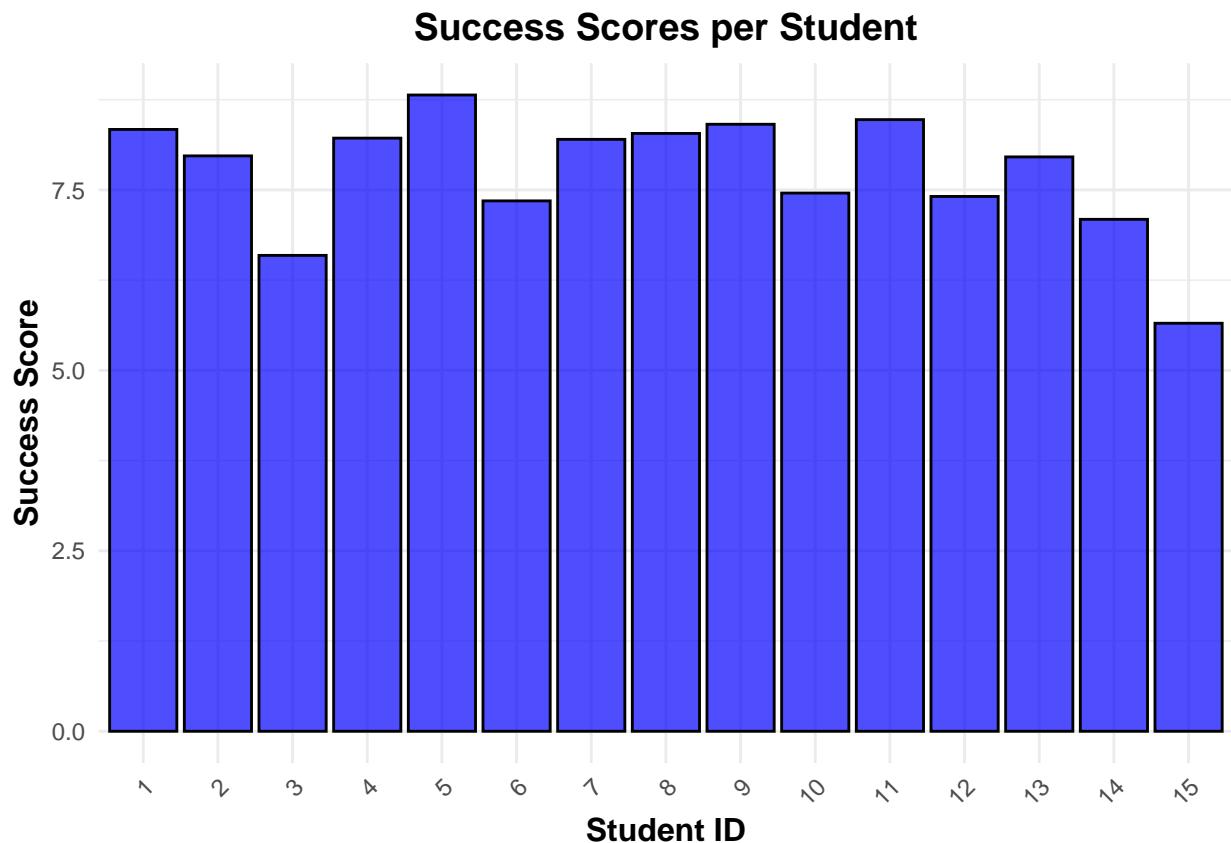
## 11 11      9.784167    8.542652    5.714000    8.473528
## 12 12      7.266671    7.720330    7.072732    7.409347
## 13 13      8.387853    7.970710    7.068622    7.957150
## 14 14      7.863167    6.445799    6.844227    7.092432
## 15 15      5.514623    5.735568    5.762224    5.652521

```

```

# Plot 1: Success scores per student
ggplot(students, aes(x = factor(ID), y = SuccessScore)) +
  geom_bar(stat = "identity", fill = "blue", color = "black", alpha = 0.7) +
  labs(
    title = "Success Scores per Student",
    x = "Student ID",
    y = "Success Score"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    axis.title = element_text(face = "bold", size = 12),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )

```



As it can be seen in the previous result, 15 students were created and assigned random scores based on the limitations that were previously set. With those randomized values, the Success score for each was calculated.

Step 2

Step 2 consists of creating the mathematical model that will solve the problem proposed. As mentioned prior, the objective function shows the student success of each student repeated for the 5 groups. Following the definition of the objective function, the constraints are also defined. As stated, there are two main constraints to be plotted, where each student can only be in one group and each group should have exactly 3 students.

```
# Number of variables: n_students * n_groups
n_vars <- n_students * n_groups

# Objective function: Maximize total success score of students assigned to groups
objective <- rep(students$SuccessScore, each = n_groups)

# Constraints:
# 1. Each student must be assigned to exactly one group
assign_constraints <- matrix(0, nrow = n_students, ncol = n_vars)
for (i in 1:n_students) {
  assign_constraints[i, ((i - 1) * n_groups + 1):(i * n_groups)] <- 1
}
assign_rhs <- rep(1, n_students) # Each student must be assigned to exactly one group

# 2. Each group must have exactly 3 students
group_constraints <- matrix(0, nrow = n_groups, ncol = n_vars)
for (j in 1:n_groups) {
  group_constraints[j, seq(j, n_vars, by = n_groups)] <- 1
}
group_rhs <- rep(3, n_groups) # Each group must have exactly 3 students

# Combine the constraints
constraints <- rbind(assign_constraints, group_constraints)

# Combine rhs and directions for the constraints
rhs <- c(assign_rhs, group_rhs)
directions <- rep("=", length(rhs))
```

Step 3

Logically, after setting the constraints and objective function for the problem, it's time to solve said problem using the lp function.

```
# Solve the optimization using lpSolve
result <- lp(
  direction = "max",
  objective.in = objective,
  const.mat = constraints,
  const.dir = directions,
  const.rhs = rhs,
  all.bin = TRUE # Ensure binary decision variables (0 or 1)
)

# Check if the solution is optimal
if (result$status == 0) {
  print("Optimal solution found!")

  # Reshape the solution vector into a matrix (n_students x n_groups)
```

```

solution_matrix <- matrix(result$solution, nrow = n_students, byrow = TRUE)
colnames(solution_matrix) <- paste("Group", 1:n_groups)

# Display the solution matrix
print("Balanced Group Assignments (Solution Matrix):")
print(solution_matrix)

# Assign groups to students based on the solution matrix
students$Group <- apply(solution_matrix, 1, function(row) which(row == 1))

# Print final student group assignments
print("Final Student Assignments (Balanced):")
print(students)

# Calculate group success scores (sum of success scores for each group)
group_summary <- aggregate(SuccessScore ~ Group, data = students, sum)
print("Balanced Group Success Scores:")
print(group_summary)

# Plot 2: Group success scores
ggplot(group_summary, aes(x = factor(Group), y = SuccessScore)) +
  geom_bar(stat = "identity", fill = "blue", alpha = 0.7) +
  labs(
    title = "Group Success Scores",
    x = "Group",
    y = "Success Score"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    axis.title = element_text(face = "bold", size = 12),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
} else {
  print("No optimal solution found.")
}

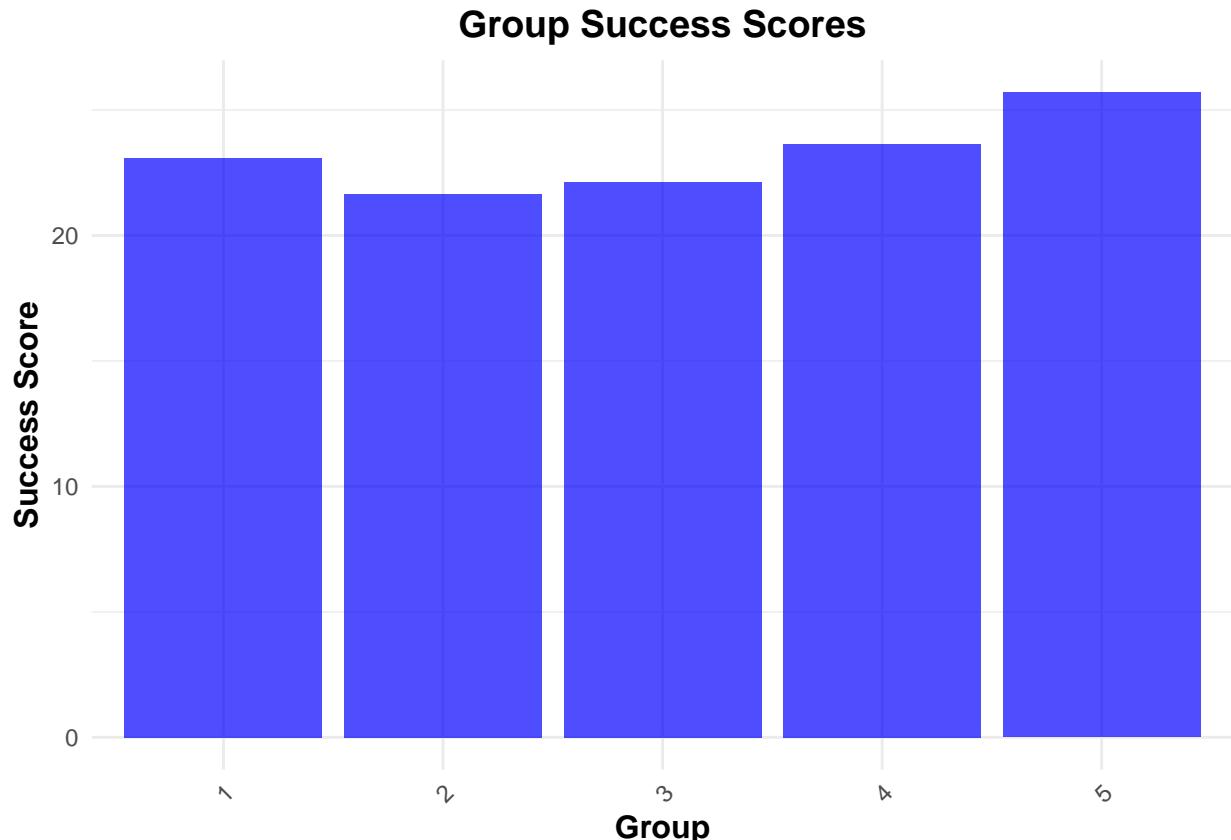
## [1] "Optimal solution found!"
## [1] "Balanced Group Assignments (Solution Matrix):"
##      Group 1 Group 2 Group 3 Group 4 Group 5
## [1,]     1     0     0     0     0
## [2,]     0     0     0     1     0
## [3,]     0     1     0     0     0
## [4,]     0     0     0     1     0
## [5,]     0     0     0     0     1
## [6,]     1     0     0     0     0
## [7,]     0     0     1     0     0
## [8,]     0     0     1     0     0
## [9,]     0     0     0     0     1
## [10,]    0     0     0     1     0
## [11,]    0     0     0     0     1
## [12,]    1     0     0     0     0
## [13,]    0     1     0     0     0

```

```

## [14,]      0      1      0      0      0
## [15,]      0      0      1      0      0
## [1] "Final Student Assignments (Balanced):"
##   ID CommunicationSkills ProblemSolving PresentationSkills SuccessScore Group
## 1  1          6.437888     9.499125     9.815121    8.337829     1
## 2  2          8.941526     6.230439     9.511495    7.971085     4
## 3  3          7.044885     5.210298     8.453526    6.592778     2
## 4  4          9.415087     6.639604     8.977337    8.217344     4
## 5  5          9.702336     9.772518     5.123068    8.814556     5
## 6  6          5.227782     9.447697     7.388980    7.347988     1
## 7  7          7.640527     8.464017     8.792298    8.200277     3
## 8  8          9.462095     8.202534     6.082040    8.282260     3
## 9  9          7.757175     9.971349     6.590905    8.409591     5
## 10 10         7.283074     8.278529     6.158129    7.456267     4
## 11 11         9.784167     8.542652     5.714000    8.473528     5
## 12 12         7.266671     7.720330     7.072732    7.409347     1
## 13 13         8.387853     7.970710     7.068622    7.957150     2
## 14 14         7.863167     6.445799     6.844227    7.092432     2
## 15 15         5.514623     5.735568     5.762224    5.652521     3
## [1] "Balanced Group Success Scores:"
##   Group SuccessScore
## 1     1     23.09516
## 2     2     21.64236
## 3     3     22.13506
## 4     4     23.64470
## 5     5     25.69767

```



We can see from these results that each student was assigned to only one group, and each group has 3 students. It can also be seen that group 5, with the weighted averages of each category, has the highest success score possible. Whilst, group 2 has the lowest. This is not indicative of the capability of the students, but their combined factors are lower than the others (for example the students from this group might be freshmen and are in the process of developing their skills).

```
group1_students <- students[students$Group == 1, ]  
print(group1_students)
```

```
##      ID CommunicationSkills ProblemSolving PresentationSkills SuccessScore Group  
## 1     1           6.437888       9.499125        9.815121    8.337829     1  
## 6     6           5.227782       9.447697        7.388980    7.347988     1  
## 12   12          7.266671       7.720330        7.072732    7.409347     1
```

We can see here an example of group formation and success score calculation. We can see Group 1 has been properly formed and their success scores properly calculated with the given weights.