MGT7180: Data Driven Decision Making
Assignment - 1

Name: Dhanush Mathighatta Shobhan Babu
Student ID: 40412492

Table Of Content

Table Of Figure

1. Problem 1

1.1Problem statement

In the context of ApTx-nova manufactures four types of make-to-order products across five plants, where certain plants are unable to manufacture specific products and each product must be produced in a single plant (which can only produce one product type), the task is to allocate plants to products in a way that maximizes the total number of batches manufactured, seeking both the optimal allocation strategy and the maximum number of batches possible.

1.2 Mathematical Model

1.2.1 Decision Variable

Let us consider x_{ij} being a binary decision variable
Having the conditions
x_{ij} = 1 if product i is being produced by plant j
x_{ij} = 0 , other wise

Product	1	2	3	4
Plant –1	1200	-	600	1000
Plant – 2	1400	1200	800	1000
Plant – 3	600	-	200	600
Plant – 4	800	-	-	1200
Plant – 5	800	400	1000	1600

From the table provided, we infer the following production capabilities for ApTx-nova's plants:
Plant-1 has the capacity to produce Product 1, Product 3, and Product 4.
Plant-2 is capable of manufacturing all four products.
Plant-3 can produce Product 1, Product 3 and Product 4, but not Product 2.
Plant-4 is equipped to produce Product 1 and Product 4, but it cannot produce Product 2 or Product 3.
Plant-5 can manufacture Product 1, Product 2, and Product 4.

1.2.2 Objective function

$\boldsymbol{Max\ z} = 1200\, x_{11} + 1400\, x_{12} + 600\, x_{13} + 800\, x_{14} + 800\, x_{15} + 0\, x_{21} + 1200\, x_{22} + 0\, x_{23} + 0\, x_{24} + 1400\, x_{25} + 600\, x_{31} + 800\, x_{32} + 200\, x_{33} + 0\, x_{34} + 1000\, x_{35} + 1000\, x_{41} + 1000\, x_{42} + 600\, x_{43} + 1200\, x_{44} + 1600\, x_{45}$

1.2.3 Constraints

Now lets us work on the given constraints

Constraint number 1 :

To ensure each product is produced by exactly one plant only :

$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = 1$
$x_{21} + x_{22} + x_{23} + x_{24} + x_{25} = 1$
$x_{31} + x_{32} + x_{33} + x_{34} + x_{35} = 1$
$x_{41} + x_{42} + x_{43} + x_{44} + x_{45} = 1$

Constraint number 2 :

To ensure each plant can produce at most one product :

$x_{11} + x_{21} + x_{31} + x_{41} \leq 1$
$x_{12} + x_{22} + x_{32} + x_{42} \leq 1$
$x_{13} + x_{23} + x_{33} + x_{43} \leq 1$
$x_{14} + x_{24} + x_{34} + x_{44} \leq 1$
$x_{15} + x_{25} + x_{35} + x_{45} \leq 1$

1.3 R Implementation

```r
#ApTx-nova - Problem - 1
#We make use of library Ompr to solve the problem
#As the required libraries are already installed we call the libraries using the syntax shown below
# Utilizing the 'ompr' library to optimize production allocation
# Load the required packages for optimization
library(tidyverse)
library(ompr) # Used for modeling the optimization problem
library(ompr.roi) # Connects 'ompr' models to the 'ROI' optimization infrastructure
library(ROI.plugin.glpk) # Enables the use of the GLPK solver through 'ROI'

# Define the optimization model
model <- MIPModel() %>%
  # Introduce binary decision variables x[i,j] where i is the product and j is the plant
  add_variable(x[i, j], i = 1:4, j = 1:5, type = "binary") %>%
  # Define the objective function to maximize the total number of batches
  set_objective(1200*x[1,1] + 1400*x[1,2] + 600*x[1,3] + 800*x[1,4] + 800*x[1,5] +
                0*x[2,1] + 1200*x[2,2] + 0*x[2,3] + 0*x[2,4] + 1400*x[2,5] +
                600*x[3,1] + 800*x[3,2] + 200*x[3,3] + 0*x[3,4] + 1000*x[3,5] +
                1000*x[4,1] + 1000*x[4,2] + 600*x[4,3] + 1200*x[4,4] + 1600*x[4,5], "max") %>%
  # Adding constraints to ensure each product is made by only one plant
  add_constraint(sum_expr(x[1,j], j = 1:5) == 1) %>%
  add_constraint(sum_expr(x[2,j], j = c(2,5)) == 1) %>%
  add_constraint(sum_expr(x[3,j], j = c(1,2,3,5)) == 1) %>%
  add_constraint(sum_expr(x[4,j], j = 1:5) == 1) %>%
  # Adding constraints to ensure that each plant produces at most one product
  add_constraint(sum_expr(x[i,1], i = 1:4) <= 1) %>%
  add_constraint(sum_expr(x[i,2], i = 1:4) <= 1) %>%
  add_constraint(sum_expr(x[i,3], i = 1:4) <= 1) %>%
  add_constraint(sum_expr(x[i,4], i = 1:4) <= 1) %>%
  add_constraint(sum_expr(x[i,5], i = 1:4) <= 1) %>%
  # Solve the model using GLPK solver and display the output
  solve_model(with_ROI(solver = "glpk", verbose = TRUE))

# Extracting the solution and displaying it
solution <- get_solution(model, x[i,j])
print(solution) # Show the optimal production allocation
model # Output a summary of the model
```

Fig 1.3 : R code implementation Problem 1

1.4 R Output

```
+   solve_model(with_ROI(solver = "glpk", verbose = TRUE))
<SOLVER MSG>  ----
GLPK Simplex Optimizer 5.0
9 rows, 20 columns, 36 non-zeros
      0: obj =  -0.000000000e+00 inf =   4.000e+00 (4)
      7: obj =   3.800000000e+03 inf =   0.000e+00 (0)
*    15: obj =   4.600000000e+03 inf =   0.000e+00 (0)
OPTIMAL LP SOLUTION FOUND
GLPK Integer Optimizer 5.0
9 rows, 20 columns, 36 non-zeros
20 integer variables, all of which are binary
Integer optimization begins...
Long-step dual simplex will be used
+    15: mip =     not found yet <=              +inf        (1; 0)
+    15: >>>>>   4.600000000e+03 <=   4.600000000e+03   0.0% (1; 0)
+    15: mip =   4.600000000e+03 <=     tree is empty   0.0% (0; 1)
INTEGER OPTIMAL SOLUTION FOUND
<!SOLVER MSG> ----
> # Extracting the solution and displaying it
```

```
<!SOLVER MSG> ----
> # Extracting the solution and displaying it
> solution <- get_solution(model, x[i,j])
> print(solution) # Show the optimal production allocation
   variable i j value
1         x 1 1     1
6         x 2 1     0
11        x 3 1     0
16        x 4 1     0
2         x 1 2     0
7         x 2 2     1
12        x 3 2     0
17        x 4 2     0
3         x 1 3     0
8         x 2 3     0
13        x 3 3     0
18        x 4 3     0
4         x 1 4     0
9         x 2 4     0
14        x 3 4     0
19        x 4 4     1
5         x 1 5     0
10        x 2 5     0
15        x 3 5     1
20        x 4 5     0
> model # Output a summary of the model
Status: success
Objective value: 4600
>
```

Fig 1.4 : R Output of Problem 1

1.5 Managerial Statement

ApTx-nova must Assign Product 1 to Plant-1, Product 2 to Plant-2, Product 3 to Plant-5, and Product 4 to Plant-4, aligning with operational constraints to reach a maximum capacity of **4,600 batches**.

Problem 2

2.1 Problem statement

Teranikx, a leading semiconductor company, has developed an innovative AI chip that outperforms competitors in both efficiency and cost. Manufactured at two locations, Fab-A in Europe and Fab-B in East Asia, the chip's production varies in capability and cost at each site. The company serves four distinct clients, each with unique demands and price points for the chips. Additionally, the cost of distributing the chips to these clients varies.

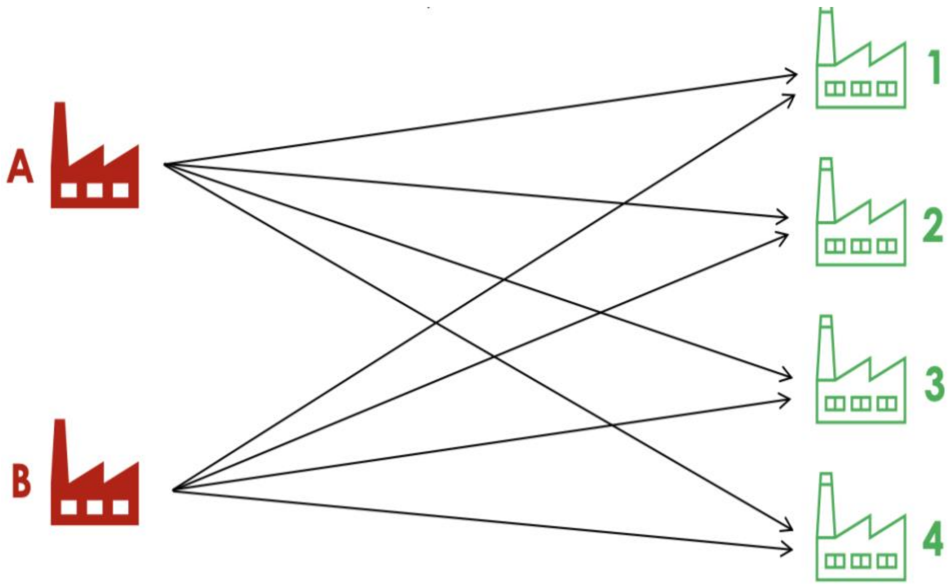
The primary question under investigation is the identification of an optimal strategy for allocating the chips to the four distinct clients in a manner that maximizes Teranikx's profit margins, while factoring in the costs associated with production and delivery. This entails calculating the precise number of chips to be distributed to each client and estimating the highest potential profit achievable through these transactions.

Fab	A	B
Production Capacity [million chips]	50	42
Production Cost [₰/chip]	1150	1250

Customer	1	2	3	4
Maximum Demand [million chips]	36	46	11	35
Offered Price [₰/chip]	1950	1850	2000	1800

Customer	1	2	3	4
Fab A	300	400	550	450
Fab B	600	300	400	250

- Fab A and B possess the ability to manufacture 50 and 42 million chips, respectively, at production costs of 1150 and 1250 per chip.
- There are four clients with demands for chips that span from 11 to 46 million, prepared to pay prices ranging from 1800 to 2000 for each chip.
- The costs to ship chips from each manufacturing site to each client differ, with transportation from Facility A costing between 300 and 550, and from Facility B, between 250 and 600 ₰ for each chip.
- An illustrated network flow chart provides a comprehensive overview of all potential distribution pathways from Facilities A and B to the four clients.



2.2 Mathematical Model

2.2.1 Decision Variables

Let us consider the Decision variable as follows:

$\boldsymbol{x_{ij}} - represent\ the\ number\ of\ chips\ sold\ to\ Customer\ i\ to\ Fab\ j$

Now based on the question let us look at the Parameters.

$\boldsymbol{P_i} - Selling\ price\ per\ chip\ for\ Customer\ i$

$\boldsymbol{C_j} - Prodution\ cost\ per\ chip\ at\ Fab\ j$

$\boldsymbol{S_{ij}} - Delivery\ cost\ per\ chip\ from\ Fab\ j\ to\ Customer\ i$

2.2.2 Objective Function:

$$\boldsymbol{Max\ z} = \sum_{i=4}^{1} \sum_{j \in \{A,B\}} (P_i - C_j - S_{ij}) x_{ij}$$

2.2.3 Constrains:

1. **Production Capacity Constrains for each FAB (in millions)**

$$\sum_{i=1}^{4} x_{iA} \leq 50$$

$$\sum_{i=1}^{4} x_{iB} \leq 42$$

2. **Demand Constrains for each Customer:**

$$x_{1A} + x_{1B} \leq 36$$
$$x_{2A} + x_{2B} \leq 42$$

$$x_{3A} + x_{3B} \leq 11$$
$$x_{4A} + x_{4B} \leq 35$$

3. **Non-Negativity and integrality:**
   $$\boldsymbol{x_{ij} \geq 0, \forall_i \in \{A,B\}, \forall_j \in \{1,2,3,4\}}$$

2.3 R implementation

```r
#AI Chip - Problem 2

#We make use of library Ompr to solve the problem
#As the required libraries are already installed we call the libraries using the syntax shown below
# Utilizing the 'ompr' library to optimize production allocation
# Load the required packages for optimization
library(ompr)
library(ompr.roi)
library(ROI.plugin.glpk) # Ensure this solver is installed

# Parameters
P_i <- c(1950, 1850, 2000, 1800) # Selling price per chip for each customer
C_j <- c(A = 1150, B = 1250) # Cost to make each chip at fab
S_ij <- matrix(c(300, 400, 550, 450, 600, 300, 400, 250), nrow = 2, byrow = TRUE) # Cost to deliver chips
prod_capacity <- c(A = 50, B = 42) # Production capacities in millions
demand <- c(36, 46, 11, 35) # Maximum demand for each customer


# Model building for question 2
model <- MIPModel() %>%
  add_variable(x[i, j], i = 1:4, j = 1:2, type = "integer", lb = 0) %>%
  set_objective(sum_expr((P_i[i] - C_j[j] - S_ij[j, i]) * x[i, j], i = 1:4, j = 1:2), "max") %>%
  add_constraint(sum_expr(x[i, 1], i = 1:4) <= prod_capacity["A"], j = 1) %>%
  add_constraint(sum_expr(x[i, 2], i = 1:4) <= prod_capacity["B"], j = 2) %>%
  add_constraint(sum_expr(x[i, j], j = 1:2) <= demand[i], i = 1:4) %>%
  solve_model(with_ROI(solver = "glpk", verbose = TRUE))



# Solution
solution <- get_solution(model, x[i, j])
cat("Solution:\n")
print(solution)
model
```

Fig 2.2 : R Code implementation of Problem 2

2.4 R Output

```
<SOLVER MSG>  ----
GLPK Simplex Optimizer 5.0
6 rows, 8 columns, 16 non-zeros
*     0: obj =  -0.000000000e+00 inf =   0.000e+00 (8)
*     4: obj =   3.535000000e+04 inf =   0.000e+00 (0)
OPTIMAL LP SOLUTION FOUND
GLPK Integer Optimizer 5.0
6 rows, 8 columns, 16 non-zeros
8 integer variables, none of which are binary
Integer optimization begins...
Long-step dual simplex will be used
+     4: mip =     not found yet <=              +inf        (1; 0)
+     4: >>>>>   3.535000000e+04 <=   3.535000000e+04   0.0% (1; 0)
+     4: mip =   3.535000000e+04 <=     tree is empty   0.0% (0; 1)
INTEGER OPTIMAL SOLUTION FOUND
<!SOLVER MSG> ----
> # Solution
> solution <- get_solution(model, x[i, j])
> cat("Solution:\n")
Solution:
> print(solution)
  variable i j value
1        x 1 1    36
3        x 2 1    14
5        x 3 1     0
7        x 4 1     0
2        x 1 2     0
4        x 2 2    31
6        x 3 2    11
8        x 4 2     0
> model
Status: success
Objective value: 35350
> 
```

Fig 2.3: R Output of Problem 2

2.5 Managerial Statement

The strategy involves allocating the manufacturing of 36 units of Product 1 to Factory 1, along with 14 units of Product 2 to the same location. It is also advised for Factory 2 to produce 31 units of Product 2, while Factory 3 should produce 11 units of Product 2. This distribution of production tasks results in a maximum profit of 35,350, adhering to our set capacity limitations and efficiently fulfilling the expected market demand.

Problem 3

3.1 Problem Statement

Apotheeker Pharmaceuticals' goal is to maximize monthly profit from producing two types of chemotherapy drugs, formulated from two base constituents with specific metrics and costs. The drugs, having fixed demand, delivery requirements, and sell prices, are influenced in their demand fulfilment by the D-metrics and P-metrics resulting from the blending process. The questions focus on determining the optimal quantities of constituents for blending to achieve maximum profit, the value of that maximum profit, and the D- and P-metrics of the drugs produced.

Types	Chemo 1	Chemo 2
Maximum D-Metric	23	23
Maximum P-Metric	88	93
Maximum Demand[vial/Month]	200,000	40,000
Minimum Delivery[vial/Month]	100,000	10,000

Constituent	EU	US
D-Metric	25	15
P-Metric	87	98
Inventory[vial]	80,000	120,000
Cost[vial]	800	1500

Table 1

- **D-Metric:** Both Chemo 1 and Chemo 2 score 23, indicating similarity in this measurement.
- **P-Metric:** Chemo 2 scores higher at 93 than Chemo 1's 88, showing a potential advantage.
- **Monthly Demand:** Chemo 1 has a significantly higher demand at 200,000 vials compared to Chemo 2's 40,000.
- **Minimum Monthly Supply:** Chemo 1 requires a minimum supply of 100,000 vials, substantially more than Chemo 2's 10,000.

Table 2

- **D-Metric Comparison:** The EU scores 25, outperforming the US's 15 in this metric.
- **P-Metric Comparison:** The US leads with a score of 98 against the EU's 87, suggesting higher performance.
- **Inventory Levels:** The US holds a larger inventory of 120,000 vials versus the EU's 80,000.
- **Cost per Vial:** Vials are priced at $800 in the EU and $1,500 in the US, indicating significant cost differences.

3.2 Mathematical Model

3.2.1 Decision Variables

Let's define the decision variables as follows.

$\boldsymbol{x1EU}$: $Quanity\ of\ Chemo\ 1\ to\ be\ blended\ for\ the\ EU\ market$

$\boldsymbol{x1US}$: $Quanity\ of\ Chemo\ 1\ to\ be\ blended\ for\ the\ US\ market$

$\boldsymbol{x2EU}$: $Quanity\ of\ Chemo\ 2\ to\ be\ blended\ for\ the\ EU\ market$

$\boldsymbol{x2EU}$: $Quanity\ of\ Chemo\ 2\ to\ be\ blended\ for\ the\ US\ market$

3.2.2 Objective function:

$\boldsymbol{Max\ z} = 1200(x1EU + x1US) + 1400(x2EU + x2US) - 800x1EU - 1500x1US - 800x2EU - 1500x2US$

3.2.3 Constraints:

1.D-metric constraints for Chemo 1 and Chemo 2:

- **For Chemo 1:** $25x1EU + 15x1US \leq 23(x1EU + x1US)$
- **For Chemo 2:** $25x2EU + 15x2US \leq 23(x2EU + x2US)$

2.P-metric constraints for Chemo 1 and Chemo 2:

- **For Chemo 1:** $87x1EU + 98x1US \geq 88(x1EU + x1US)$
- **For Chemo 2:** $87x2EU + 98x2US \geq 93(x2EU + x2US)$

3.Demand Constraints

- $\boldsymbol{x1EU + x1US \geq 100000}$
- $\boldsymbol{x2EU + x2US \geq 10000}$

4.Supply Constraints

- $x1EU + x1US \leq 200000$
- $x2EU + x2US \leq 40000$

5.Inventory Constraints

- $x1EU + x2EU \leq 80000$
- $x1US + x2US \leq 120000$

3.3 R Implementation

1_404124... * | Assignement_1_Problem_2_40412... * | Assigment_1_Problem_3_4041249... * | Untitled1* | Untitled

Source on Save | Run | Source

```r
# Load necessary libraries for modeling and solving MIP
library(ompr)           # For modeling
library(ompr.roi)       # For solving the model using ROI (R Optimization Infrastructure)
library(ROI.plugin.glpk) # GLPK (GNU Linear Programming Kit) solver plugin

# Model setup
model <- MIPModel() %>%
  # Decision variables: quantities of EU and US base constituents for Chemo1 and Chemo2
  add_variable(x1EU, type = "integer", lb = 0) %>%
  add_variable(x1US, type = "integer", lb = 0) %>%
  add_variable(x2EU, type = "integer", lb = 0) %>%
  add_variable(x2US, type = "integer", lb = 0) %>%

  # Objective: Maximize profit calculated as revenue minus cost for both drugs
  set_objective(1200 * (x1EU + x1US) + 1400 * (x2EU + x2US) - 800 * x1EU - 1500 * x1US - 800 * x2EU - 1500 * x2

  # Constraints
  # D-metric constraints for Chemo1 and Chemo2
  add_constraint(25*x1EU + 15*x1US <= 23 * (x1EU + x1US)) %>%
  add_constraint(25*x2EU + 15*x2US <= 23 * (x2EU + x2US)) %>%

  # P-metric constraints for Chemo1 and Chemo2
  add_constraint(87*x1EU + 98*x1US >= 88 * (x1EU + x1US)) %>%
  add_constraint(87*x2EU + 98*x2US >= 93 * (x2EU + x2US)) %>%

  # Demand constraints to meet market requirements
  add_constraint(x1EU + x1US >= 100000) %>%
  add_constraint(x1EU + x1US <= 200000) %>%
  add_constraint(x2EU + x2US >= 10000) %>%
  add_constraint(x2EU + x2US <= 40000) %>%

  # Supply constraints based on available base constituents
  add_constraint(x1EU + x2EU <= 80000) %>%
  add_constraint(x1US + x2US <= 120000)

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

57:1 (Top Level) R Script


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


```r

# Extract solution values for decision variables
x1EU_sol <- get_solution(solution, x1EU)
x1US_sol <- get_solution(solution, x1US)
x2EU_sol <- get_solution(solution, x2EU)
x2US_sol <- get_solution(solution, x2US)

# Calculate D-metrics and P-metrics for Chemo1 and Chemo2 based on solution
D_metric_Chemo1 <- (25*x1EU_sol + 15*x1US_sol) / (x1EU_sol + x1US_sol)
D_metric_Chemo2 <- (25*x2EU_sol + 15*x2US_sol) / (x2EU_sol + x2US_sol)
P_metric_Chemo1 <- (87*x1EU_sol + 98*x1US_sol) / (x1EU_sol + x1US_sol)
P_metric_Chemo2 <- (87*x2EU_sol + 98*x2US_sol) / (x2EU_sol + x2US_sol)

# Output the results: optimal quantities, profit, and drug metrics
cat("Quantities to blend for Chemo1: EU =", x1EU_sol, ", US =", x1US_sol, "\n")
cat("Quantities to blend for Chemo2: EU =", x2EU_sol, ", US =", x2US_sol, "\n")
cat("Maximum monthly profit: €", solution$objective_value, "\n")
cat("D-metrics: Chemo1 =", D_metric_Chemo1, ", Chemo2 =", D_metric_Chemo2, "\n")
cat("P-metrics: Chemo1 =", P_metric_Chemo1, ", Chemo2 =", P_metric_Chemo2, "\n")
```

57:1 (Top Level) R Script

Fig 3.3 : R Code implementation of Problem 3

3.4 R output

```
> # Solve the model using the GLPK solver
> solution <- solve_model(model, with_ROI(solver = "glpk"))
> # Extract solution values for decision variables
> x1EU_sol <- get_solution(solution, x1EU)
> x1US_sol <- get_solution(solution, x1US)
> x2EU_sol <- get_solution(solution, x2EU)
> x2US_sol <- get_solution(solution, x2US)
> # Calculate D-metrics and P-metrics for Chemo1 and Chemo2 based on solution
> D_metric_Chemo1 <- (25*x1EU_sol + 15*x1US_sol) / (x1EU_sol + x1US_sol)
> D_metric_Chemo2 <- (25*x2EU_sol + 15*x2US_sol) / (x2EU_sol + x2US_sol)
> P_metric_Chemo1 <- (87*x1EU_sol + 98*x1US_sol) / (x1EU_sol + x1US_sol)
> P_metric_Chemo2 <- (87*x2EU_sol + 98*x2US_sol) / (x2EU_sol + x2US_sol)
> # Output the results: optimal quantities, profit, and drug metrics
> cat("Quantities to blend for Chemo1: EU =", x1EU_sol, ", US =", x1US_sol, "\n")
Quantities to blend for Chemo1: EU = 75455 , US = 24545
> cat("Quantities to blend for Chemo2: EU =", x2EU_sol, ", US =", x2US_sol, "\n")
Quantities to blend for Chemo2: EU = 4545 , US = 5455
> cat("Maximum monthly profit: €", solution$objective_value, "\n")
Maximum monthly profit: € 2.5e+07
> cat("D-metrics: Chemo1 =", D_metric_Chemo1, ", Chemo2 =", D_metric_Chemo2, "\n")
D-metrics: Chemo1 = 22.5455 , Chemo2 = 19.545
> cat("P-metrics: Chemo1 =", P_metric_Chemo1, ", Chemo2 =", P_metric_Chemo2, "\n")
P-metrics: Chemo1 = 89.69995 , Chemo2 = 93.0005
> 
```

Fig 3.4: R Output of Problem 2

3.5 Managerial Statement

The analysis advises Apotheeker Pharmaceuticals to produce 75,454 Chemo1 vials with EU materials and 24,545 with US materials, and for Chemo2, 4,545 vials from the EU and 5,455 from the US, targeting a €25 million profit. It aligns with D-metrics, Chemo1 at 22.5455 and Chemo2 at 19.545, and P-metrics, Chemo1 at 89.69995 and Chemo2 at 93.0005, ensuring cost-effectiveness, quality compliance, and market demand fulfilment to improve market position.