HW solutions

Md Rasheduzzaman

2025-05-07

dataframe, matrices, list, factor, vector, etc.

Table of contents

1 L3: Matrices and Lists

1.1 Task 1: Protein Concentration in Samples

We measured the concentration (in µg/µL) of three proteins (P1, P2, P3) in four samples (S1–S4):

1.1.1 1. Matrices

```
# Making Protein Matrix
ProteinMatrix <- matrix(
  c(5, 3, 2,
    7, 6, 4),
  nrow = 2, byrow = TRUE
)
rownames(ProteinMatrix) = c("Sample1", "Sample2")
colnames(ProteinMatrix) = c("ProteinX", "ProteinY", "ProteinZ")
ProteinMatrix
```

```
        ProteinX ProteinY ProteinZ
Sample1        5        3        2
Sample2        7        6        4
```

Now goes the weight matrix

```
# Making weight matrix
WeightVector <- matrix(
  c(0.5, 1.0, 1.5),
  nrow=3, byrow = TRUE
)
rownames(WeightVector) = c("ProteinX", "ProteinY", "ProteinZ")
colnames(WeightVector) = c("Weight")
WeightVector
```

```
         Weight
ProteinX    0.5
ProteinY    1.0
ProteinZ    1.5
```

Now, multiply them.

```r
# Multiplying Matrices
TotalConc = ProteinMatrix %*% WeightVector
colnames(TotalConc) <- "Total_Protein_Conc"
print(TotalConc)
```

```
        Total_Protein_Conc
Sample1                8.5
Sample2               15.5
```

1.1.2 2. Making transpose of ProteinMatrix

```r
ProteinMatTranspose = t(ProteinMatrix)
ProteinMatTranspose
```

```
         Sample1 Sample2
ProteinX       5       7
ProteinY       3       6
ProteinZ       2       4
```

1.1.3 3. Identity Matrix

```r
I <- diag(3)
Identitycheck = ProteinMatrix %*% I
colnames(Identitycheck) <- c("ProteinX", "ProteinY", "ProteinZ")
Identitycheck
```

```
        ProteinX ProteinY ProteinZ
Sample1        5        3        2
Sample2        7        6        4
```

1.1.4 4.1. Total Protein per Sample

```r
rowSums(ProteinMatrix)
```

```
Sample1 Sample2
     10      17
```

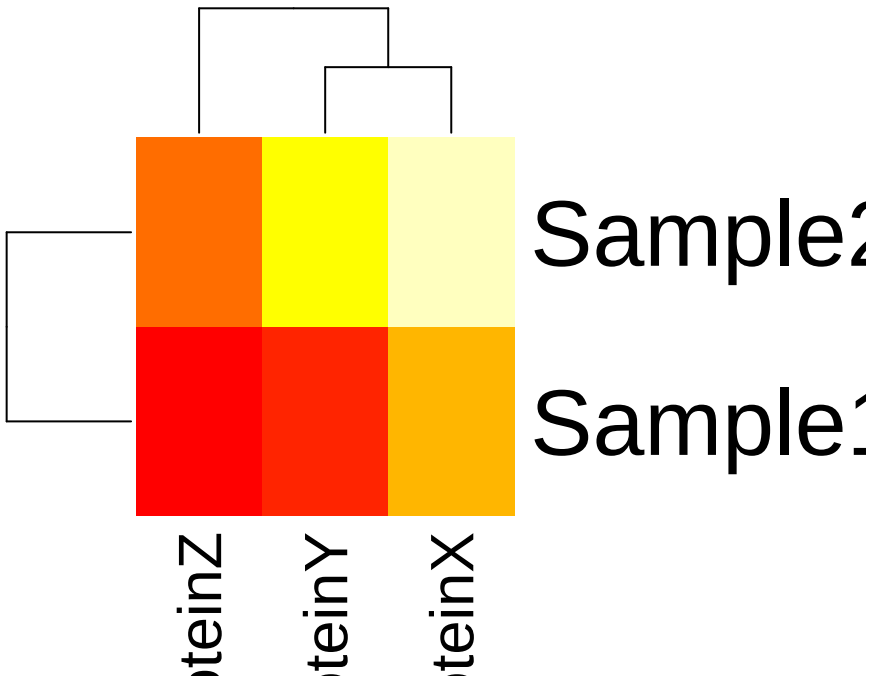
1.1.5 4.2. Total Protein per Protein Type

```
colSums(ProteinMatrix)
```

```
ProteinX ProteinY ProteinZ
      12        9        6
```

1.1.6 4.3. Heatmap of Protein Concentrations

```
heatmap(ProteinMatrix, scale = "none", col = heat.colors(10))
```



1.1.7 Interpretation

- Multiplying the protein levels by the weight vector shows how much each protein contributes in a sample. The result shows total protein concentration per sample.
- The result shows that sample S2 has the highest protein burden.
- The identity matrix represents no protein interactions or measurement biases. It is a simple matrix calculation.
- New calculation:

```r
# changing the weight of ProteinZ to 3.0
newweightvector = matrix(
  c(0.5, 1.0, 3.0),
  nrow=3, byrow = TRUE
)
rownames(WeightVector) = c("ProteinX", "ProteinY", "ProteinZ")
colnames(WeightVector) = c("Weight")
newTotalconc = ProteinMatrix %*% newweightvector
colnames(newTotalconc) <- "Total_Protein_Conc"
newTotalconc
```

```
        Total_Protein_Conc
Sample1               11.5
Sample2               21.5
```

Still, S2 has more protein burden.

Bonus:

- Heatmap reveals PX is most abundant across all samples.

1.2 Task 2: Gene-to-Protein Translation

```r
# making  Gene Expression matrix
GeneExpression <- matrix(
  c(10, 8, 5,
    15, 12, 10),
  nrow = 2, byrow = TRUE
)
rownames(GeneExpression) <- c("Sample1", "Sample2")
colnames(GeneExpression) <- c("GeneA", "GeneB", "GeneC")
GeneExpression
```

```
        GeneA GeneB GeneC
Sample1    10     8     5
Sample2    15    12    10
```

Translation efficiency:

```r
# making Translation Matrix
TranslationMatrix <- matrix(
  c(1.5, 0 , 0,
  0, 1.2, 0,
  0, 0, 1.8),
nrow = 3, byrow = TRUE
)

rownames(TranslationMatrix) <- c("GeneA", "GeneB", "GeneC")
TranslationMatrix
```

```
      [,1] [,2] [,3]
GeneA  1.5  0.0  0.0
GeneB  0.0  1.2  0.0
GeneC  0.0  0.0  1.8
```

1.2.1 Protein Output

```r
# computing Protein matrix
Protein_matrix <- GeneExpression %*% TranslationMatrix
print(Protein_matrix)
```

```
        [,1] [,2] [,3]
Sample1 15.0  9.6    9
Sample2 22.5 14.4   18
```

1.2.2 2. Transpose

```r
# Transpose of GeneExpression matrix
GeneExpression_Transpose <- t(GeneExpression)
GeneExpression_Transpose
```

```
      Sample1 Sample2
GeneA      10      15
GeneB       8      12
GeneC       5      10
```

The new matrix represnts a matrix where the rows and columns of GeneExpression matrix have been interchanged.

1.2.3 3. Identity matrix and multipliccation

```
# Creating Identity matrix
I <- diag(3)
I
```

```
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```

Now, multiply:

```
Product_matrix = TranslationMatrix %*%  I
Product_matrix
```

```
      [,1] [,2] [,3]
GeneA  1.5  0.0  0.0
GeneB  0.0  1.2  0.0
GeneC  0.0  0.0  1.8
```

The product is identical to TranslationMatrix

1.2.4 4. Sub-matrix:

```
# making submatrix A
A = matrix(
  c(10, 8,
    15, 12), nrow=2, byrow = TRUE
)
rownames(A) = c("sample1", "sample2")
colnames(A) = c("GeneA", "GeneB")

A
```

```
        GeneA GeneB
sample1    10     8
sample2    15    12
```

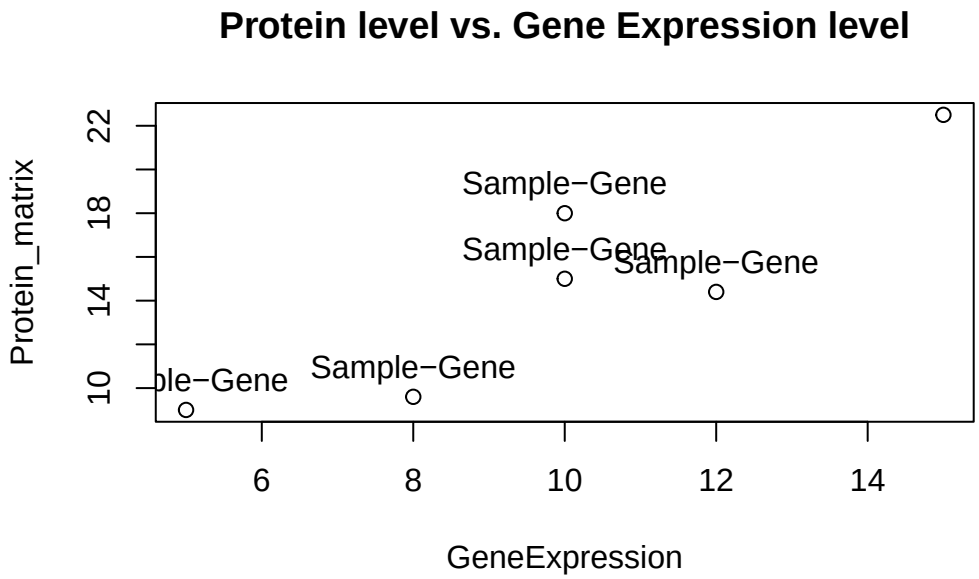
```
# finding inverse of A
#inv_A <- solve(A)
#inv_A
```

The inverse matrix could not be calculated since A is a singular matrix. So, A * A^-1 is also not possible.

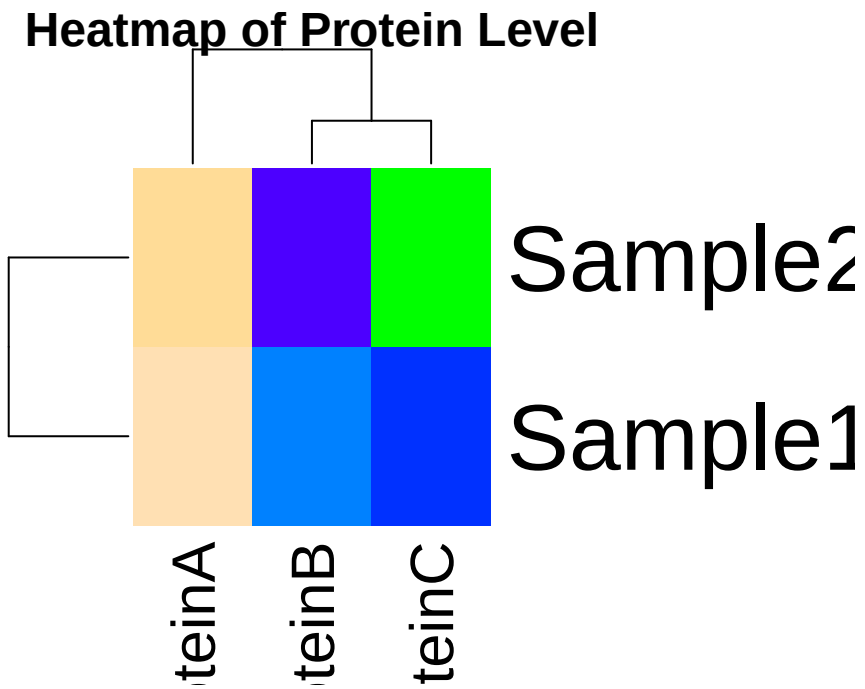
1.2.5 Visualization tasks:

- 1. MARplot

```
# generating MARplot-style scatter plot
plot(GeneExpression, Protein_matrix, type="p", main="Protein level vs. Gene Expression lev
labels <- "Sample-Gene"
text(GeneExpression, Protein_matrix, labels = labels, pos=3)
```

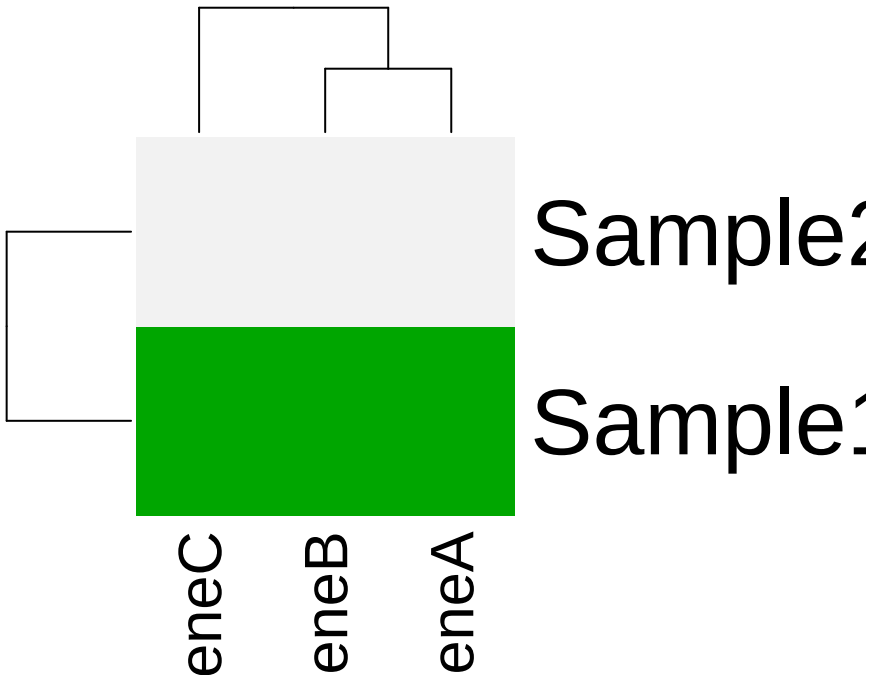



```
# generating a heatmap
heatmap(Protein_matrix, main= "Heatmap of Protein Level", Rowv = TRUE, Colv = TRUE, labRow
```



- 2. Heatmap of Expression:

```
heatmap(GeneExpression, col = terrain.colors(10), scale = "column")
```



1.2.6 Interpretation

1. Matrix multiplication allows each gene in both samples to be multiplied to their respective translation efficiency. So, the product shows how successfully each gene is translated")
2. The diagonal TranslationMatrix make sense biologically because they show translation efficiency of each gene and there is no other interaction between them. Although there could be interaction in real-world scenarios.
3. If Sample2 has higher protein levels even with similar gene expression, it means that more mRNAs are translated to proteins compared to Sample1"
4. The upward trend in MARplot may indicate an increase in translation efficacy and downward trend may indicate a decline in translation efficacy"
5. Clustering in the heatmap may suggest which samples are most similar to each other based on their prot.

1.3 Task 3: Animal Breeding – Bull Ranking by Economic Traits

Define Data

```
# Define Bull EBVs
BullEBVs <- matrix(c(
  400, 1.2, 0.8,
  500, 1.5, 0.6
), nrow = 2, byrow = TRUE)

rownames(BullEBVs) <- c("Bull1", "Bull2")
colnames(BullEBVs) <- c("Milk_yield", "Growth_rate", "Fertility")

# Define Economic Weights
EconomicWeights <- matrix(c(0.002, 50, 100), ncol = 1)
```

1 Compute Total Economic Value

```
TotalValue <- BullEBVs %*% EconomicWeights
TotalValue
```

```
       [,1]
Bull1 140.8
Bull2 136.0
```

Interpretation

Bull1: $(400 \times 0.002) + (1.2 \times 50) + (0.8 \times 100) = 140.8$ Bull2: $(500 \times 0.002) + (1.5 \times 50) + (0.6 \times 100) = 136.0$

Bull1 is more valuable economically.

2 Biological Interpretation

Economic weights convert genetic merit (EBVs) into economic merit. Traits with higher financial importance have a larger impact, regardless of absolute EBV values.

3 Multiply with Identity Matrix

```
I3 <- diag(3)
BullEBVs_identity <- BullEBVs %*% I3
BullEBVs_identity
```

```
      [,1] [,2] [,3]
Bull1  400  1.2  0.8
Bull2  500  1.5  0.6
```

Interpretation

Multiplying by identity matrix returns the original matrix. It confirms that EBV structure is preserved.

4 Remove Milk Yield and Recalculate Total Value

```
BullEBVs_noMilk <- BullEBVs[, -1]
EconomicWeights_noMilk <- EconomicWeights[2:3, , drop = FALSE]

TotalValue_noMilk <- BullEBVs_noMilk %*% EconomicWeights_noMilk
TotalValue_noMilk
```

```
      [,1]
Bull1  140
Bull2  135
```

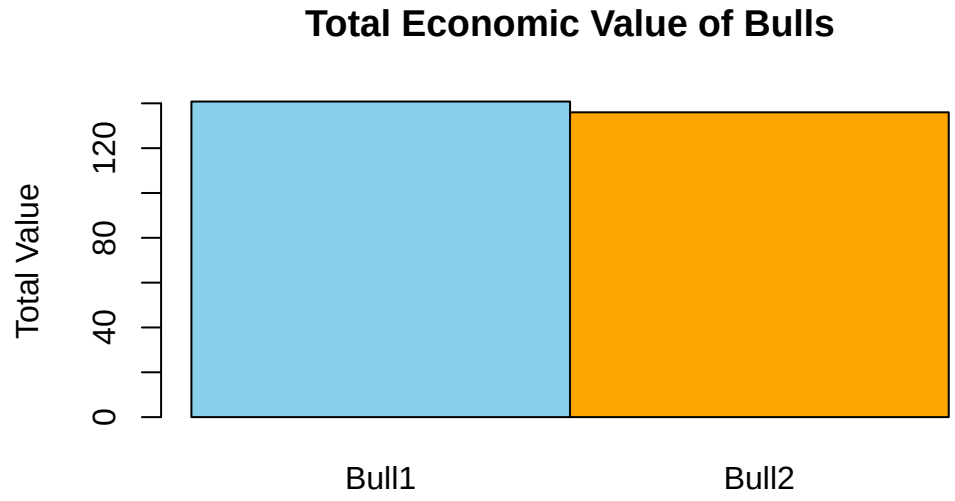
Interpretation

Bull1: $(1.2 \times 50) + (0.8 \times 100) = 140$ Bull2: $(1.5 \times 50) + (0.6 \times 100) = 135$

Bull1 still ranks higher, but by a smaller margin.

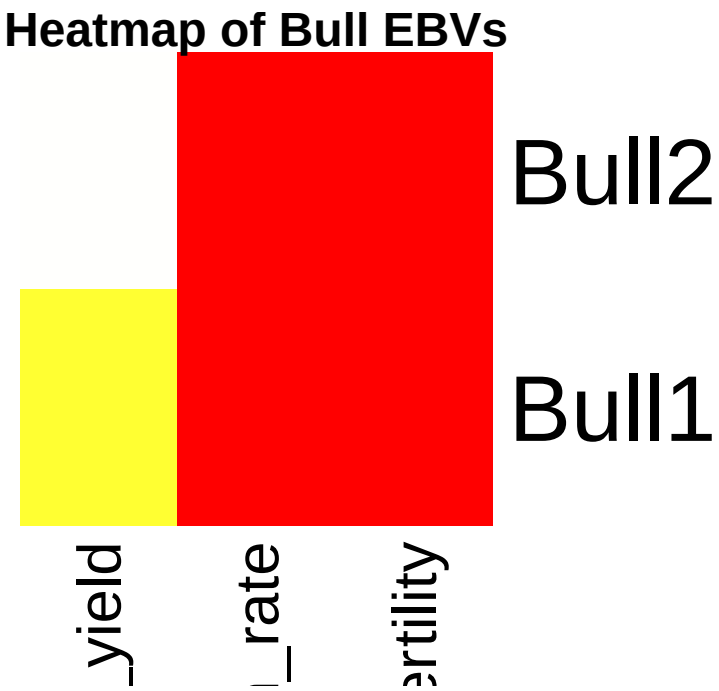
5 Bar Plot: Total Economic Value (Base R)

```r
barplot(
  TotalValue,
  beside = TRUE,
  names.arg = rownames(BullEBVs),
  col = c("skyblue", "orange"),
  main = "Total Economic Value of Bulls",
  ylab = "Total Value"
)
```



6 Heatmap of EBVs (Base R)

```r
heatmap(
  BullEBVs,
  Rowv = NA,
  Colv = NA,
  scale = "none",
  col = heat.colors(256),
  main = "Heatmap of Bull EBVs"
)
```



Interpretation Questions

How do economic weights affect trait importance?

Traits with higher weights contribute more to the total economic value.This makes them more influential in ranking and selection.

Why might you ignore milk yield?

Milk yield may be excluded in systems focusing on fertility, growth, or when it is no longer a limiting factor.Environmental or economic contexts may also shift trait priorities.

What is the value of heatmaps?

Heatmaps visually compare EBVs across bulls and traits.They help detect patterns, outliers, and clusters easily in multivariate data.

Can this method be extended to more bulls and traits?

Yes. This method scales to any number of bulls or traits.Just ensure the EBVs matrix and economic weights are dimensionally compatible.

1.4 Task 4: Plant Breeding – Trait Contributions from Parental Lines

```r
# making parent traits matrix
ParentTraits = matrix(
  c(7, 5, 3,
    6, 8, 4,
    5, 6, 6), nrow=3, byrow= TRUE
)
rownames(ParentTraits) = c("P1", "P2", "P3")
colnames(ParentTraits) = c("T1 Drought resistance", "T2 Yield", "T3 Maturation time")
print(ParentTraits)
```

```
   T1 Drought resistance T2 Yield T3 Maturation time
P1                     7        5                  3
P2                     6        8                  4
P3                     5        6                  6
```

```r
# making HybridWeights matrix
HybridWeights = matrix(
  c(0.5, 0.3, 0.2), nrow=3, byrow= TRUE
)
print("HybridWeights:")
```

```
[1] "HybridWeights:"
```

```r
print(HybridWeights)
```

```
     [,1]
[1,]  0.5
[2,]  0.3
[3,]  0.2
```

```r
# Transpose of HybridWeights
HybridWeightsTranspose = t(HybridWeights)
print("HybridWeightsTranspose:")
```

```
[1] "HybridWeightsTranspose:"
```

```
print(HybridWeightsTranspose)
```

```
     [,1] [,2] [,3]
[1,]  0.5  0.3  0.2
```

1.4.1 computing Hybrid traits matrix

```
HybridTraits =  HybridWeightsTranspose %*% ParentTraits
print("The HybridTraits matrix is:")
```

```
[1] "The HybridTraits matrix is:"
```

```
print(HybridTraits)
```

```
     T1 Drought resistance T2 Yield T3 Maturation time
[1,]                   6.3      6.1                3.9
```

```
# Explaining what it means biologically when one parent contributes more to a particular t
print("A parent contributing more to a particular trait than others makes it more desirabl
```

```
[1] "A parent contributing more to a particular trait than others makes it more desirable an
```

```
# making identity matrix and multiplying with ParentTraits
I = diag(3)
print(I)
```

```
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```

```
Product = I %*% ParentTraits
print(Product)
```

```
     T1 Drought resistance T2 Yield T3 Maturation time
[1,]                     7        5                  3
[2,]                     6        8                  4
[3,]                     5        6                  6
```

```r
print("Observation:")
```

```
[1] "Observation:"
```

```r
print(" I × ParentTraits is identical to ParentTraits matrix")
```

```
[1] " I × ParentTraits is identical to ParentTraits matrix"
```

```r
print("I × ParentTraits represents the original matrix and therefore no transformation or
```

```
[1] "I × ParentTraits represents the original matrix and therefore no transformation or chang
```

```r
# subsetting
parentsub = ParentTraits[, 1:2]
print("Subset matrix of ParentTraits")
```

```
[1] "Subset matrix of ParentTraits"
```

```r
print(parentsub)
```

```
   T1 Drought resistance T2 Yield
P1                     7        5
P2                     6        8
P3                     5        6
```

```r
newhybridtraits = HybridWeightsTranspose %*% parentsub
print("recalculated hybrid traits is")
```

```
[1] "recalculated hybrid traits is"
```

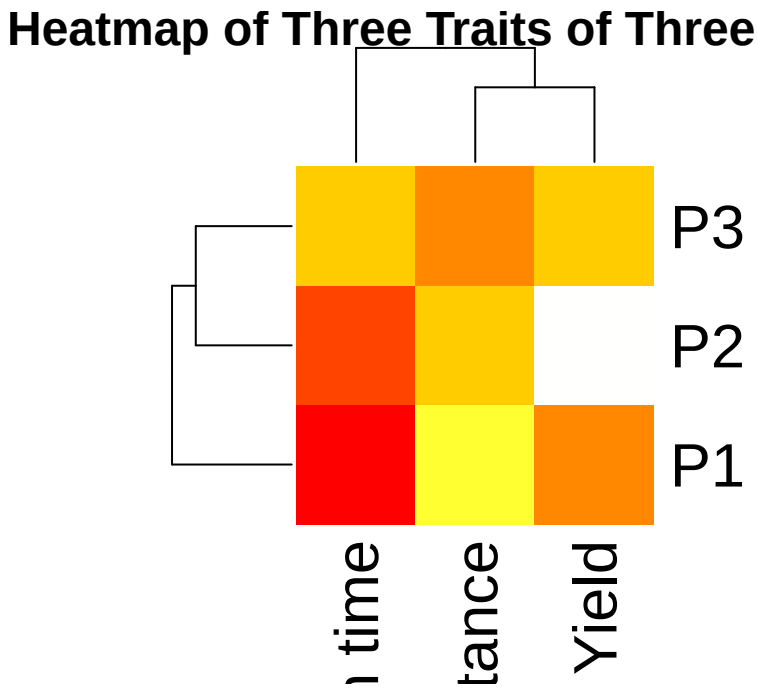
```
print(newhybridtraits)
```

```
     T1 Drought resistance T2 Yield
[1,]                   6.3      6.1
```

Removing maturation trait does not provide insight about the maturation time of the three parent plants. The new matrix has one less column but the values of other two traits is unchanged.

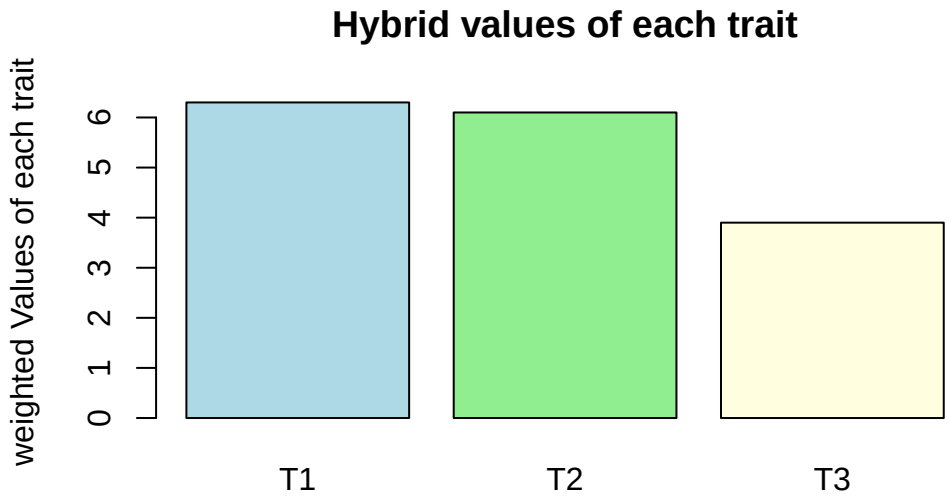
```
## Visulaization tasks

# A heatmap of ParentTraits matrix
heatmap(ParentTraits, Rowv=TRUE, Colv=TRUE, labRow= rownames(ParentTraits), labCol= colnam
```




```
# A barplot of HybridTraits
hybridtraitsvec = c(T1=6.3, T2=6.1, T3=3.9)


barplot(hybridtraitsvec, main="Hybrid values of each trait", ylab="weighted Values of each
```




```
## Interpretation Questions

print("The parent traits with more weight might affect hybrid's performance more.")
```

```
[1] "The parent traits with more weight might affect hybrid's performance more."
```

```
print("I think the identity matrix serves as a control which is used to compare the matrix
```

```
[1] "I think the identity matrix serves as a control which is used to compare the matrix with
```

```
equal_weights = matrix(c(0.3, 0.3, 0.3), nrow=1, byrow=TRUE)
change_in_hybrid_traits = equal_weights %*% ParentTraits
print("The change in hybrid weights will be")
```

```
[1] "The change in hybrid weights will be"
```

```
print(change_in_hybrid_traits)
```

```
     T1 Drought resistance T2 Yield T3 Maturation time
[1,]                   5.4      5.7                3.9
```

```r
print("I think this simple model ignores the influence of various abiotic and biotic facto
```

```
[1] "I think this simple model ignores the influence of various abiotic and biotic factors su
```

1.5 Task 5: Lists

```r
# Making a MasterList
bioList = list(
ProteinConc = list(ProteinMatrix, WeightVector),
ProteinMap = list(GeneExpression, TranslationMatrix),
Plant = list(ParentTraits, HybridWeights),
Animal = list(BullEBVs, EconomicWeights)
)

print(bioList)
```

```
$ProteinConc
$ProteinConc[[1]]
        ProteinX ProteinY ProteinZ
Sample1        5        3        2
Sample2        7        6        4

$ProteinConc[[2]]
         Weight
ProteinX    0.5
ProteinY    1.0
ProteinZ    1.5


$ProteinMap
$ProteinMap[[1]]
        GeneA GeneB GeneC
Sample1    10     8     5
Sample2    15    12    10

$ProteinMap[[2]]
      [,1] [,2] [,3]
GeneA  1.5  0.0  0.0
GeneB  0.0  1.2  0.0
GeneC  0.0  0.0  1.8
```

```
$Plant
$Plant[[1]]
   T1 Drought resistance T2 Yield T3 Maturation time
P1                     7        5                  3
P2                     6        8                  4
P3                     5        6                  6

$Plant[[2]]
     [,1]
[1,]  0.5
[2,]  0.3
[3,]  0.2


$Animal
$Animal[[1]]
      Milk_yield Growth_rate Fertility
Bull1        400         1.2       0.8
Bull2        500         1.5       0.6

$Animal[[2]]
      [,1]
[1,] 2e-03
[2,] 5e+01
[3,] 1e+02
```

```
# Task1
print("List of top level components:")
```

```
[1] "List of top level components:"
```

```
print("1. ProteinConc 2. ProteinMap 3. Plant 4. Animal")
```

```
[1] "1. ProteinConc 2. ProteinMap 3. Plant 4. Animal"
```

```
print("List of nested components:")
```

```
[1] "List of nested components:"
```

```
print("1. ProteinMatrix 2. WeightVector 3. GeneExpression 4. TranslationMatrix 5. ParentTr
```

```
[1] "1. ProteinMatrix 2. WeightVector 3. GeneExpression 4. TranslationMatrix 5. ParentTraits
```

```
# Task 2
bioList[[2]][[1]]
```

```
        GeneA GeneB GeneC
Sample1    10     8     5
Sample2    15    12    10
```

```
bioList[[2]][[2]]
```

```
      [,1] [,2] [,3]
GeneA  1.5  0.0  0.0
GeneB  0.0  1.2  0.0
GeneC  0.0  0.0  1.8
```

```
bioList[[4]][[1]]
```

```
      Milk_yield Growth_rate Fertility
Bull1        400         1.2       0.8
Bull2        500         1.5       0.6
```

```
bioList[[4]][[2]]
```

```
      [,1]
[1,] 2e-03
[2,] 5e+01
[3,] 1e+02
```

```
matrix_of_plant_entry = bioList[[3]][[1]]
print(matrix_of_plant_entry)
```

```
   T1 Drought resistance T2 Yield T3 Maturation time
P1                     7        5                  3
P2                     6        8                  4
P3                     5        6                  6
```

```
weight_for_protein_conc = bioList[[1]][[2]]
print(weight_for_protein_conc)
```

```
         Weight
ProteinX    0.5
ProteinY    1.0
ProteinZ    1.5
```

```
# Task 3
Weighted_gene_expression_score = bioList[[1]][[1]] %*% bioList[[1]][[2]]
print(Weighted_gene_expression_score)
```

```
        Weight
Sample1    8.5
Sample2   15.5
```

```
Contribution_of_transcripts_to_each_protein = bioList[[2]][[1]] %*% bioList[[2]][[2]]
print(Contribution_of_transcripts_to_each_protein)
```

```
        [,1] [,2] [,3]
Sample1 15.0  9.6    9
Sample2 22.5 14.4   18
```

```
Hybrid_trait_values = t(bioList[[3]][[2]]) %*% bioList[[3]][[1]]
print(Hybrid_trait_values)
```

```
     T1 Drought resistance T2 Yield T3 Maturation time
[1,]                   6.3      6.1                3.9
```

```
Bull_total_economic_value = bioList[[4]][[1]] %*% bioList[[4]][[2]]
print(Bull_total_economic_value)
```

```
       [,1]
Bull1 140.8
Bull2 136.0
```

```r
# Task 4

subset_BullEBVs = bioList[[4]][[1]][, 2:3]
print(subset_BullEBVs)
```

```
      Growth_rate Fertility
Bull1         1.2       0.8
Bull2         1.5       0.6
```

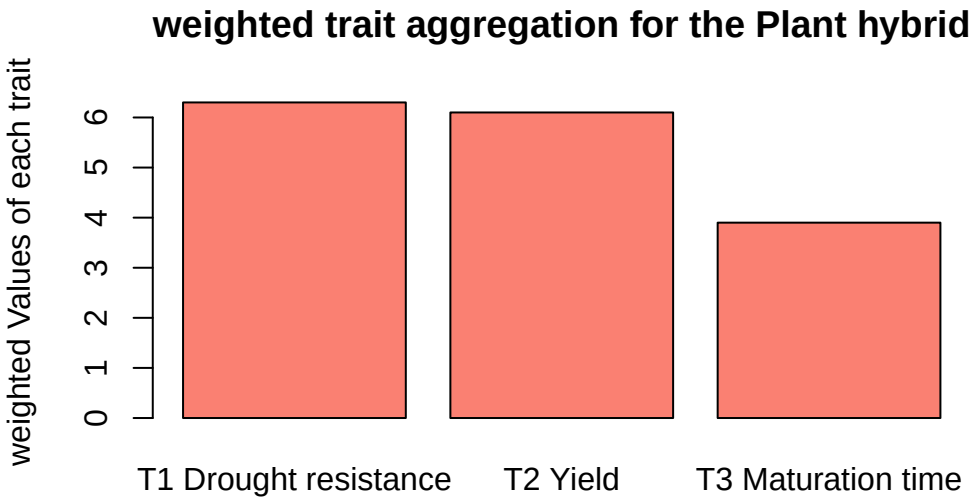
```r
Ecoweightsub = bioList[[4]][[2]][2:3, ]
print(Ecoweightsub)
```

```
[1]  50 100
```

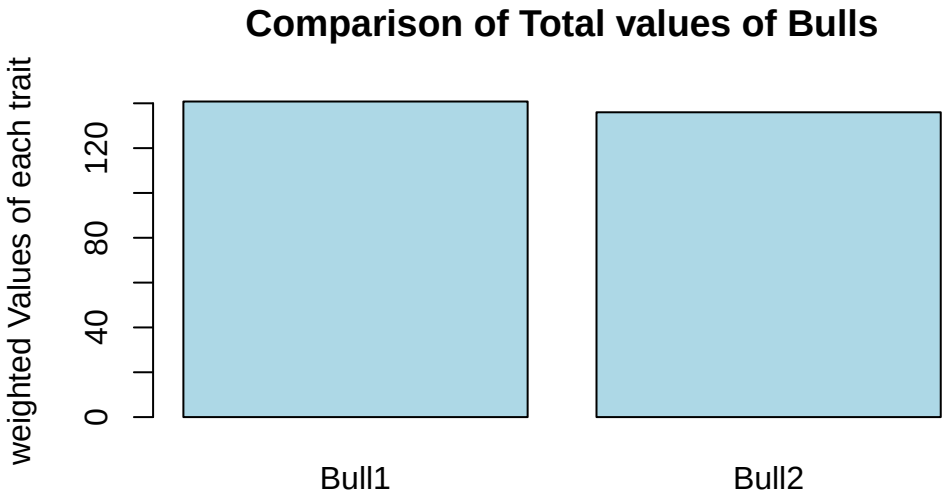
```r
Recalculated_total_value = subset_BullEBVs %*% Ecoweightsub
print(Recalculated_total_value)
```

```
      [,1]
Bull1  140
Bull2  135
```

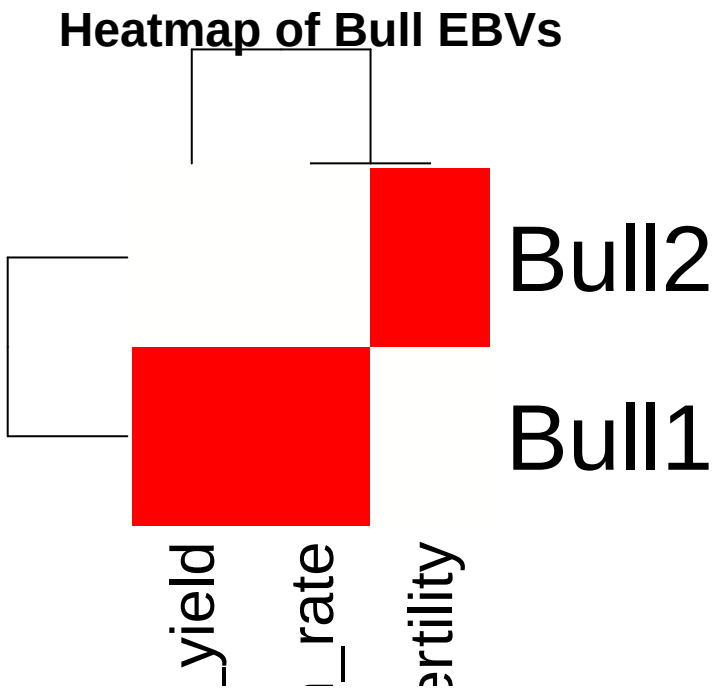
```r
## Visualization Tasks
# Creating barplot showing the result of weighted trait aggregation for the Plant hybrid
barplot(Hybrid_trait_values, main="weighted trait aggregation for the Plant hybrid", ylab=
```




```
# Creating barplot showing the total breeding values for each bull
barplot(t(Bull_total_economic_value), main="Comparison of Total values of Bulls", ylab="we
```




```
# Creating heatmap
heatmap(bioList[[4]][[1]], Rowv=TRUE, Colv=TRUE, labRow= rownames(bioList[[4]][[1]]), labC
```




```
## Interpretation Questions
print("Structuring data using a list help to logically organise different datatypes such a
```

```
[1] "Structuring data using a list help to logically organise different datatypes such as mat
```

```
print("Nested lists require multiple levels of indexing where there is a higher chance of
```

```
[1] "Nested lists require multiple levels of indexing where there is a higher chance of makin
```

```
print("Yes, this structure could be scaled for real datasets with many samples or traits")
```

```
[1] "Yes, this structure could be scaled for real datasets with many samples or traits"
```

```
print("How would you loop over all elements in bioList to apply the same function? Not sur
```

```
[1] "How would you loop over all elements in bioList to apply the same function? Not sure"
```

```
print("How can this list structure be useful for building automated bioinformatics pipelin
```

```
[1] "How can this list structure be useful for building automated bioinformatics pipelines? I
```