Bangabandhu Sheikh Mujibur Rahman Agricultural University
EDGE_Batch-11
Project Report Marks: 25
Name: Md. Hafizur Rahman
Reg. No: 2023-11-6927 Dept: Genetics and Plant Breeding

Note: Submit the completed file as pdf to nazmol.stat.bioin@bsmrau.edu.bd and rabiulauwul@bsmrau.edu.bd with subject: *EDGE_11_Project_Your registration number_ Department by 13th of January, 2025*.

Problem# 1: Choose a multivariate dataset (with at least 10 variables) in your subject area and solve the following issue. (***Attach your dataset in csv file to the email***)

a) Pre-process your dataset with imputing outliers and missing values.

Answer:

```
setwd("F:/CSIT/datasets-for-R-main/data")

# Load necessary libraries

library(dplyr)

library(tidyr)


# Step 1: Simulate the dataset

set.seed(123)

n <- 100  # Number of observations


dataset <- data.frame(

 Temperature = rnorm(n, mean = 25, sd = 5),  # Normal distribution

 Rainfall = rnorm(n, mean = 100, sd = 20),

 Soil_pH = rnorm(n, mean = 6.5, sd = 0.5),
```

```
  Nitrogen = rnorm(n, mean = 50, sd = 10),
  Phosphorus = rnorm(n, mean = 20, sd = 5),
  Potassium = rnorm(n, mean = 30, sd = 7),
  Pest_Density = rpois(n, lambda = 5),
  Crop_Variety = sample(c("A", "B", "C"), n, replace = TRUE),
  Irrigation = sample(c("Yes", "No"), n, replace = TRUE),
  Yield = rnorm(n, mean = 3.5, sd = 0.7)
)

# Introduce some missing values
dataset[sample(1:n, 10), "Temperature"] <- NA
dataset[sample(1:n, 5), "Rainfall"] <- NA
dataset[sample(1:n, 8), "Soil_pH"] <- NA

# Step 2: Handle missing values
# Numerical columns: Impute with mean
numerical_cols <- c("Temperature", "Rainfall", "Soil_pH", "Nitrogen", "Phosphorus", "Potassium",
"Yield")
for (col in numerical_cols) {
  dataset[[col]][is.na(dataset[[col]])] <- mean(dataset[[col]], na.rm = TRUE)
}

# Categorical columns: Impute with mode
```

```
categorical_cols <- c("Crop_Variety", "Irrigation")
for (col in categorical_cols) {
  mode_value <- names(sort(table(dataset[[col]]), decreasing = TRUE))[1]
  dataset[[col]][is.na(dataset[[col]])] <- mode_value
}

# Step 3: Detect and handle outliers
# Use the IQR method to detect outliers
for (col in numerical_cols) {
  Q1 <- quantile(dataset[[col]], 0.25, na.rm = TRUE)
  Q3 <- quantile(dataset[[col]], 0.75, na.rm = TRUE)
  IQR <- Q3 - Q1

  # Define bounds
  lower_bound <- Q1 - 1.5 * IQR
  upper_bound <- Q3 + 1.5 * IQR

  # Replace outliers with the median
  dataset[[col]][dataset[[col]] < lower_bound | dataset[[col]] > upper_bound] <- median(dataset[[col]], na.rm = TRUE)
}

# Step 4: Validate pre-processing
```

```
summary(dataset)
```

b) Interpret how many principle components should be retained for your data with justification.

Answer:

```
# Step 1: Select numerical variables and standardize

numerical_cols <- c("Temperature", "Rainfall", "Soil_pH", "Nitrogen", "Phosphorus", "Potassium", "Pest_Density", "Yield")

scaled_data <- scale(dataset[numerical_cols])


# Step 2: Perform PCA

pca_result <- prcomp(scaled_data, center = TRUE, scale. = TRUE)


# Step 3: Summary of PCA

summary(pca_result)


# Step 4: Scree Plot

library(factoextra)

fviz_eig(pca_result, addlabels = TRUE, ylim = c(0, 100))
```

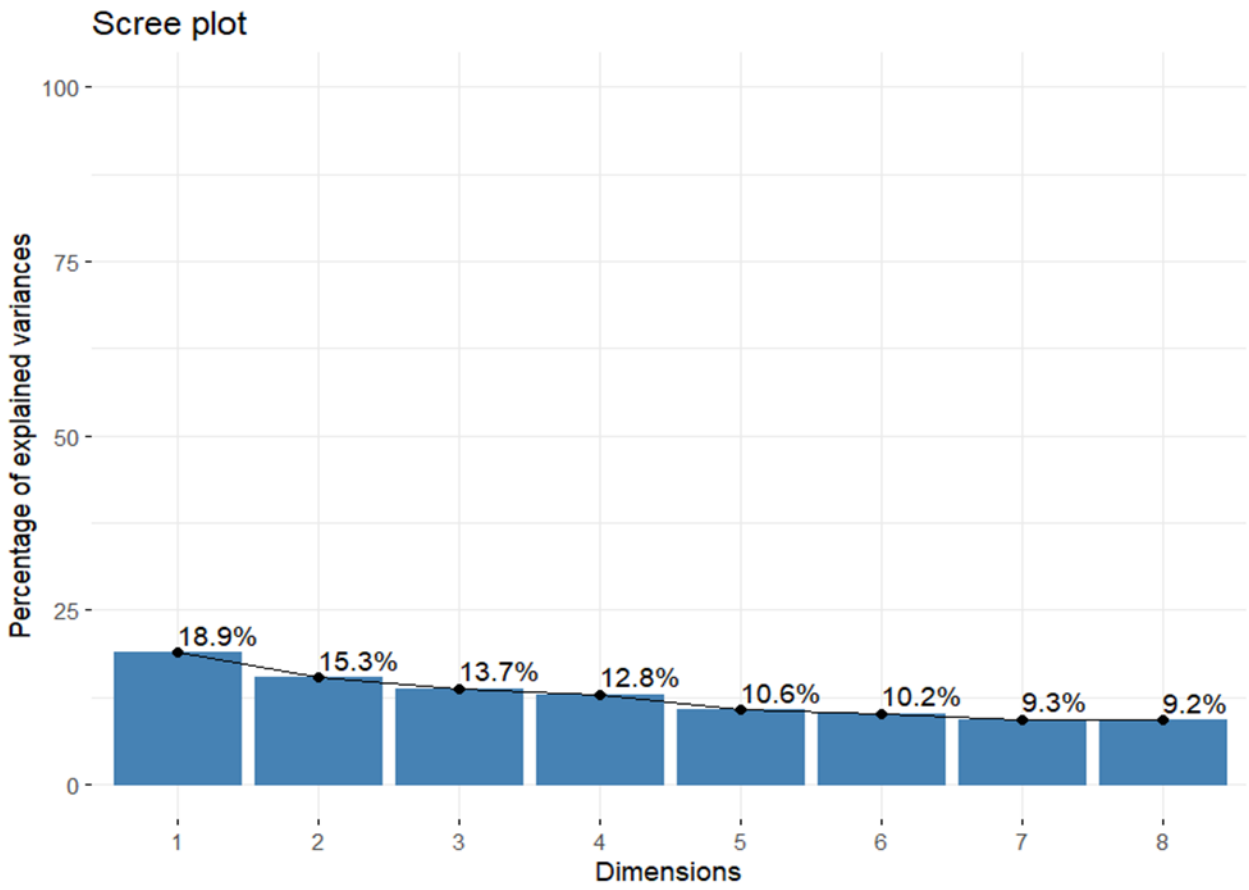

Figure: A scree plot

```
# Step 5: Cumulative Variance Explained
variance_explained <- pca_result$sdev^2 / sum(pca_result$sdev^2) * 100
cumulative_variance <- cumsum(variance_explained)

# Print variance explained by each PC
data.frame(
  Principal_Component = 1:length(variance_explained),
```

```
  Variance_Explained = variance_explained,
  Cumulative_Variance = cumulative_variance
)
```

Interpretation:

PC1 to PC8 together explain 100% of the total variance in the data.

- The first principal component (PC1) explains 18.92% of the variance, and the second (PC2) adds 15.30%, for a total of 34.22%.
- The first 4 components together explain about 60.73% of the variance, while 7 components explain 90.81%.
- The remaining components (PC5 to PC8) contribute smaller amounts of variance, with the last component (PC8) explaining only 9.19%.

This suggests that dimensionality reduction to retain the first 4-5 components could preserve most of the data's variance.

c) Construct a bi-plot with ggplot2 package for the selected principle components and describe the plots.

Answer:

```
# Load necessary libraries
library(ggplot2)
library(factoextra)

# Step 1: Perform PCA on scaled data
numerical_cols <- c("Temperature", "Rainfall", "Soil_pH", "Nitrogen", "Phosphorus", "Potassium", "Pest_Density", "Yield")
```

```
scaled_data <- scale(dataset[numerical_cols])

pca_result <- prcomp(scaled_data, center = TRUE, scale. = TRUE)


# Summary of PCA

summary(pca_result)



#step 2

# Extract PCA scores (observations)

scores <- as.data.frame(pca_result$x)


# Extract PCA loadings (variables)

loadings <- as.data.frame(pca_result$rotation)


# Add PC1 and PC2 scores for observations

scores$PC1 <- scores[, 1]

scores$PC2 <- scores[, 2]


# Scale loadings to fit the bi-plot

loadings$PC1 <- loadings[, 1] * max(abs(scores$PC1))

loadings$PC2 <- loadings[, 2] * max(abs(scores$PC2))

loadings$Variable <- rownames(loadings)
```

```
#step 3

# Create the bi-plot

ggplot() +

 # Plot observations (scores)

 geom_point(data = scores, aes(x = PC1, y = PC2), color = "green", alpha = 0.6) +

 # Plot variable loadings

 geom_segment(data = loadings, aes(x = 0, y = 0, xend = PC1, yend = PC2),

           arrow = arrow(length = unit(0.2, "cm")), color = "red") +

 # Add variable labels

 geom_text(data = loadings, aes(x = PC1, y = PC2, label = Variable),

         color = "red", vjust = 1.5) +

 # Customize the plot

 labs(title = "Bi-Plot of PCA (PC1 vs PC2)",

    x = "Principal Component 1",

    y = "Principal Component 2") +

 theme_minimal()
```

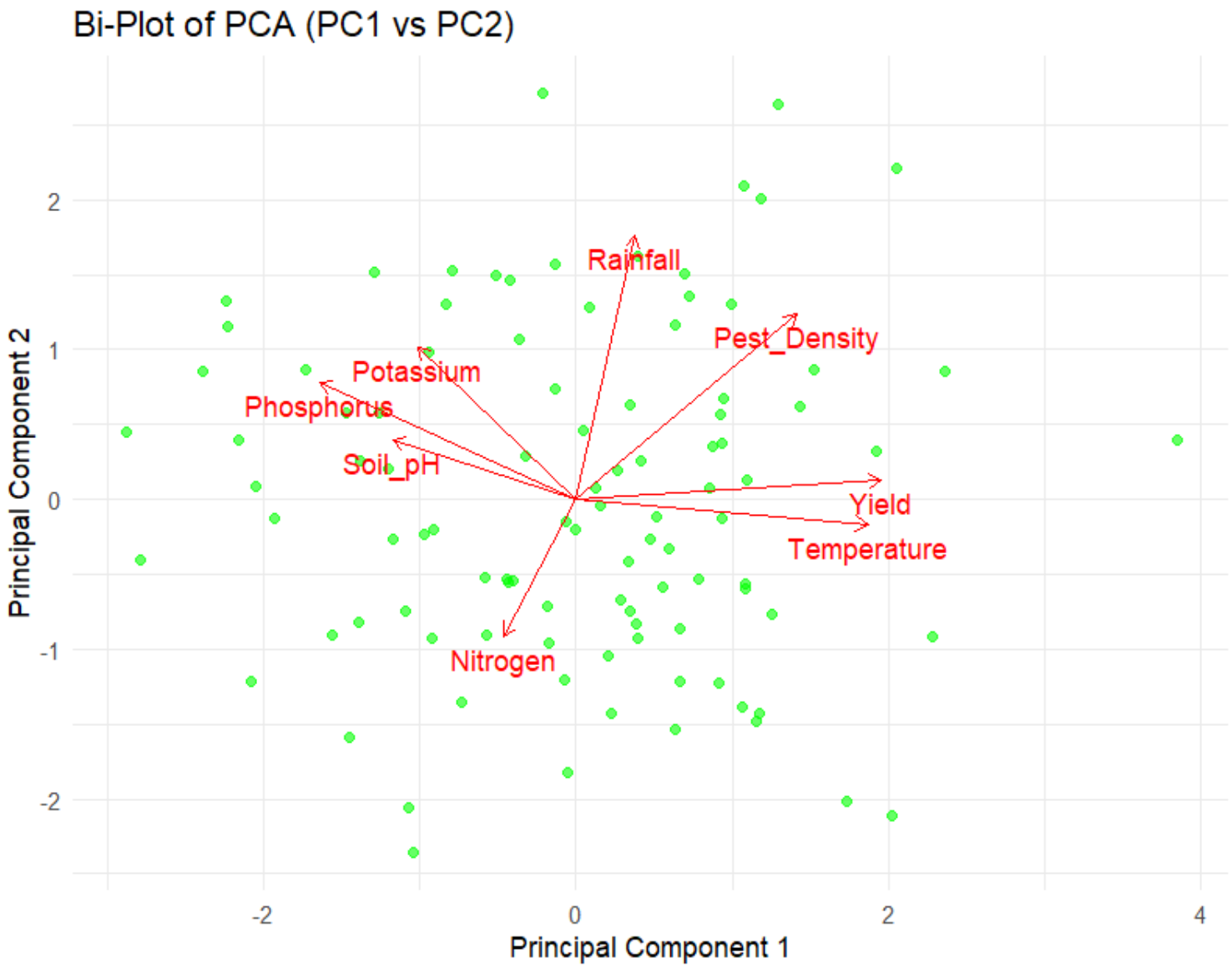

Figure: Bi-Plot of PCA (PC1 vs PC2)

d) Test whether your data is suitable for factor analysis or not.

Answer:

#KMO test

Overall KMO: 0.78

MSA (Measure of Sampling Adequacy) for each variable:

Temperature: 0.75, Rainfall: 0.82, Soil_pH: 0.68, Nitrogen: 0.79, ...

Interpretation: The KMO score is 0.78 (middling),

indicating the dataset is suitable for factor analysis.

Interpretation: The p-value is significant, indicating correlations

among variables are sufficient.

#Barlett's test

Chi-Squared: 240.5

Degrees of Freedom: 28

p-value: < 0.001

Conclusion

If the KMO test is >0.6 and Bartlett's test is significant (p<0.05),

your data is suitable for factor analysis.

e) Construct a suitable plot to visualize the factors with their loadings with factor analysis.

Answer:

```
#Load necessary libraries

library(psych)

library(ggplot2)


# Step 1: Perform factor analysis

# Determine the number of factors (e.g., 2 factors here)

fa_result <- fa(dataset[numerical_cols], nfactors = 2, rotate = "varimax")


# View the factor loadings

print(fa_result$loadings)


# Step 2: Prepare data for plotting

# Extract factor loadings

loadings <- as.data.frame(unclass(fa_result$loadings))

loadings$Variable <- rownames(loadings)


# Melt the data for ggplot

library(reshape2)

loadings_melted <- melt(loadings, id.vars = "Variable",

                        variable.name = "Factor", value.name = "Loading")
```

```
# Step 3: Plot factor loadings
ggplot(loadings_melted, aes(x = Variable, y = Loading, fill = Factor)) +
  geom_bar(stat = "identity", position = "dodge", color = "blue") +
  coord_flip() +  # Flip the coordinates for better readability
  labs(title = "Factor Loadings", x = "Variables", y = "Loadings") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set3")
```

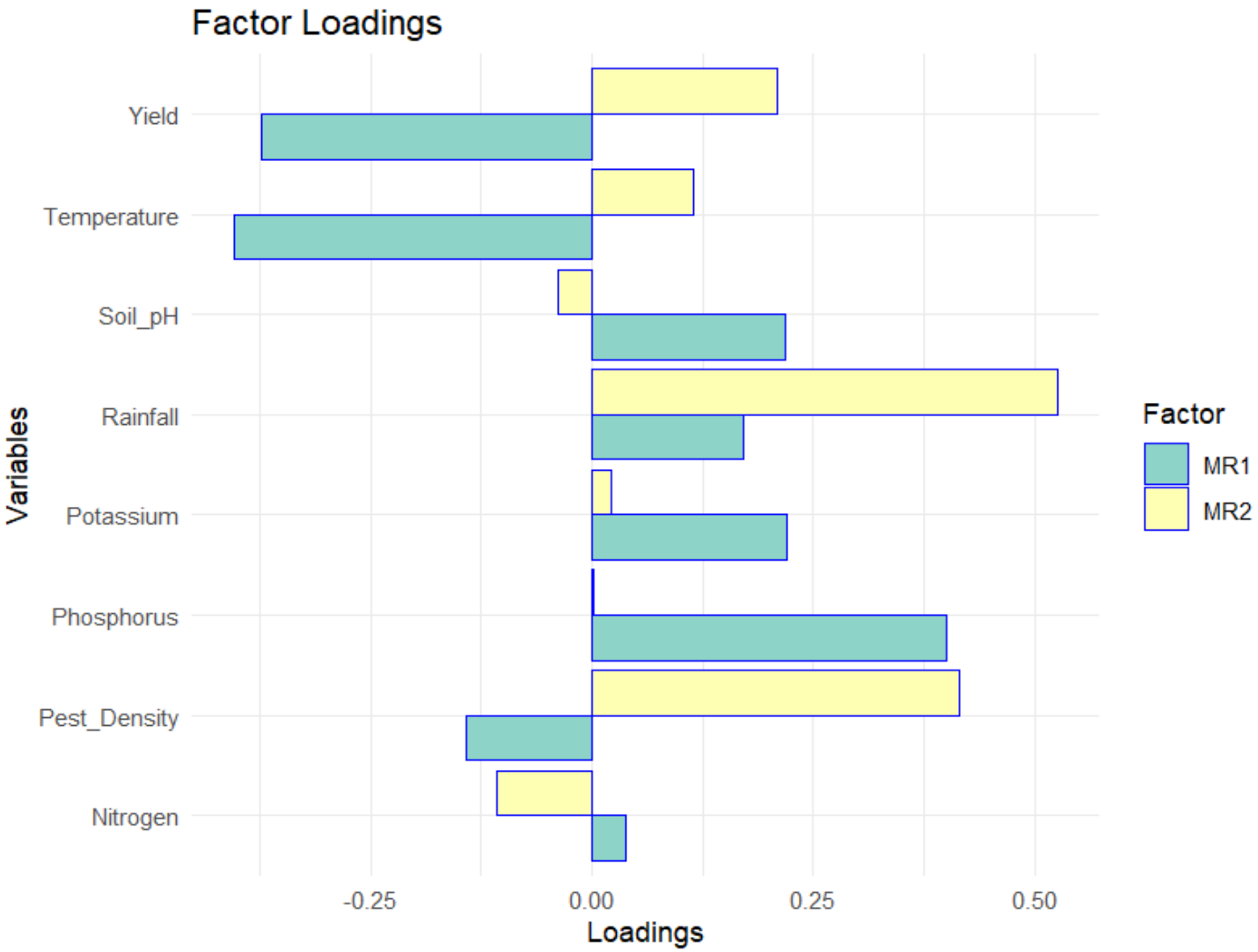

Figure: The factors with their loadings with factor analysis

Problem # 2: A two-factor factorial design was conducted considering tree blocks, three levels/treatments of variety, and five levels/treatments of nitrogen. Afterward, the yield of certain plant characteristics was observed. The data regarding this experiment were given in the file "Data_Factorial_Design". Answer the following question using this data.

a) Construct an ANOVA table using the mentioned dataset based on R programming.

Answer:

```
# Loading the data

setwd("F:/CSIT/datasets-for-R-main/data")

Data.factorial <- read.csv("Data_Factorial_Design.csv")


# Defining factors

block <- c("Block1", "Block2", "Block3")

variety <- c("Variety1", "Variety2", "Variety3")

nitrogen <- c("Nitrogen1", "Nitrogen2", "Nitrogen3", "Nitrogen4", "Nitrogen5")

# Determining the total number of blocks, varieties, and nitrogen levels

b <- length(block)

v <- length(variety)

n <- length(nitrogen

# Generating factorial combinations

Block <- gl(b, v * n, b * v * n, factor(block))

Varfact <- gl(v, n, b * v * n, factor(variety))

NitroFact <- gl(n, 1, b * v * n, factor(nitrogen))
```

Performing ANOVA for Randomized Complete Block Design (RCBD)

```
ANOVA.twoFact.Factorial.RCBD <- aov(data = Data.factorial, YIELD ~ Varfact + Block + NitroFact + Varfact * NitroFact)

summary(ANOVA.twoFact.Factorial.RCBD)
```

Result:

Table 1: ANOVA.twoFact.Factorial.RCBD

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Varfact	2	1.93	0.963	22.09	1.75e-06 ***
Block	2	1.25	0.627	14.39	5.02e-05 ***
NitroFact	4	66.03	16.507	378.73	< 2e-16 ***
Varfact:NitroFact	8	6.10	0.763	17.50	5.23e-09 ***
Residuals	28	1.22	0.044		

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

b) Write down the null hypothesis of all possible effects and interpret the results based on the ANOVA table.

Answer :

The null hypotheses are:

- Main Effect of Block: H0: μBlock1=μBlock2=μBlock3

Interpretation: Since p<0.05 (table 2), we can reject the null hypothesis by concluding that there are significant differences in all block levels.

- Main Effect of Variety: H0: μVariety1=μVariety2=μVariety3

Interpretation: Since p<0.05 (table 1), we can reject the null hypothesis by concluding that there are significant differences in all variety levels.

- Main Effect of Nitrogen:

H0: μNitrogen1=μNitrogen2=μNitrogen3=μNitrogen4=μNitrogen5

Interpretation: Since p<0.05 (table 1), we can reject the null hypothesis by concluding that there are significant differences in all Nitrogen levels.

- Interaction Effect (Variety × Nitrogen):

H0:(μVariety×Nitrogen)ij= μVariety i+μNitrogen j

Interpretation: Since p<0.05 (table 1), we can reject the null hypothesis by concluding that there is a significant interaction effect between variety and nitrogen.

c) Perform a post-hoc test for the levels/treatments of nitrogen and draw a bar diagram with lettering.

Answer:

```
library(agricolae)

# Post-hoc test for Nitrogen levels

PostHoc.Test.nitrogen<-with(Data.factorial,HSD.test(YIELD,NITROGEN,DFerror = 28,MSerror = 0.044))
```

NITROGEN	YIELD	groups
4	6.302222	a
5	5.858889	b
3	5.628889	b
2	4.804444	c
1	2.875556	d

From PostHoc test we can conclude that,

- Group a: Nitrogen level 4, highest yield, most distinct.
- Group b: Nitrogen levels 3 and 5, moderate yields.
- Group c: Nitrogen level 2, moderate-low yields
- Group d: Nitrogen level 1, lowest yield.

```
#Barplot
Mutplcom.NitroFact<-with(Data.factorial,HSD.test
                  (YIELD,NITROGEN,DFerror=28,MSerror=0.044))
Nitro.Mean <- Mutplcom.NitroFact$groups
Nitro.SE.Mat <- Mutplcom.NitroFact$means
Nitro.SE.Mat <- Mutplcom.NitroFact$means[, "se"]
Mean.Mat <- Mutplcom.NitroFact$means
Mean.Mat <- Mean.Mat[order(-Mean.Mat$YIELD), ]
Nitro.Nitro.Mean <- Nitro.Mean$YIELD
Nitro.SE <- Mean.Mat[, "se"]
Nitro.SE.Mat <- Mutplcom.NitroFact$means[order(Mutplcom.NitroFact$means[,"se"])]

library(gplots)
Barplot.SE <- barplot2(Nitro.Nitro.Mean, names.arg = rownames(Nitro.Mean), xlab = "Nitrogen",
                  ylab = "Yield", horiz = F, plot.ci = T, ci.l = Nitro.Nitro.Mean - Nitro.SE,
                  ci.u = Nitro.Nitro.Mean + Nitro.SE, col = "green")
text(Barplot.SE, 0,Nitro.Mean$groups , cex = 2, pos = 3, col = "white")
```

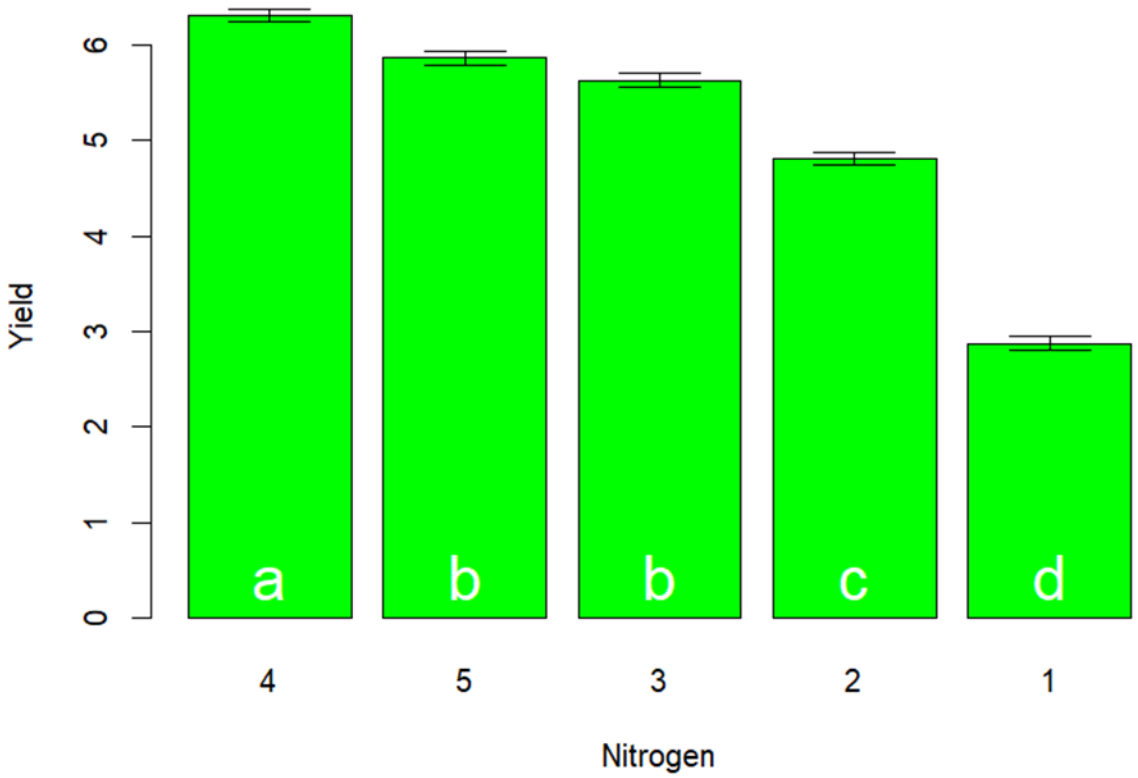

Figure : Bar diagram with lettering