Bangabandhu Sheikh Mujibur Rahman Agricultural University
EDGE_Batch-06
Project Report Marks: 25
Name: Roksana Ferdous Tama
Reg. No: 2019-05-4961 Dept. of Biotechnology and Genetic Engineering

Note: Submit the completed file as *pdf* to nazmol.stat.bioin@bsmrau.edu.bd with subject *EDGE_06_Project_Your registration number_ Department by 26th of December, 2024*.

Problem# 1:

A split-plot design was conducted considering tree blocks, three levels/treatments of variety in the main plot, and five levels/treatments of nitrogen in the split-plot. Afterward, the yield of certain plant characteristics was observed. The data regarding this experiment were given in the file "Split_Plot_Design". Answer the following question using this data.

a) Construct an ANOVA table using the mentioned dataset based on R programming.
b) Write down the null hypothesis of all possible effects and interpret the results based on the ANOVA table.
c) Perform a post-hoc test for the interaction effect (variety × nitrogen) and draw a bar diagram with lettering.

Problem# 2:

a) What is principal component analysis?
b) What are the main purposes of principle component analysis in your study area?
c) Compute the eigenvalue and eigenvector using the iris data based on R programming.
d) Construct a scree plot and interpret how many principal components should be retained to interpret the iris dataset.
e) Construct a bi-plot for the iris data based on R programming and interpret the results.

ANSWER:

Solution 01:

a) **Construction of an ANOVA table using the mentioned dataset based on R programming is given below:**

 # Code

```
data<-read.csv("Split_Plot_Design.csv")

attach(data)

dim(data)


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

variety<-c("variety1","variety2","variety3")

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


b<-length(blk)

v<-length(variety)

n<-length(nitrogen)


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

vari.fact<-gl(v,n,b*v*n,factor(variety))

nitro.fact<-gl(n,1,b*v*n,factor(nitrogen))


library(agricolae)


ANOVA.Fact<-aov(YIELD~vari.fact+nitro.fact+block+vari.fact*nitro.fact,data = data)

summary(ANOVA.Fact)
```

Result:

```
Df Sum Sq Mean Sq F value   Pr(>F)
vari.fact            2   1.93   0.963   22.09 1.75e-06 ***
nitro.fact           4  66.03  16.507  378.73  < 2e-16 ***
block                2   1.25   0.627   14.39 5.02e-05 ***
vari.fact:nitro.fact 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) The null hypothesis of all possible effects and interpretion of the results based on the ANOVA table is given below:

Variety (vari.fact):

- **Null Hypothesis (H0):** The mean yield does not differ among varieties.
- **Result:** With $p=1.75\times10-6$ we reject H0. Variety significantly affects yield.

Nitrogen (nitro.fact):

- **Null Hypothesis (H0):** The mean yield does not differ among nitrogen levels.
- **Result:** With $p<2\times10-16$, we reject H0. Nitrogen levels have a highly significant impact on yield.

Block:

- **Null Hypothesis (H0):** Yield does not vary due to tree blocks.
- **Result:** With $p=5.02\times10-5$ we reject H0. Tree blocks significantly affect yield.

Interaction (vari.fact:nitro.fact):

- **Null Hypothesis (H0):** There is no interaction effect between variety and nitrogen on yield.
- **Result:** With $p=5.23\times10-9$ we reject H0. A significant interaction exists, meaning the effect of nitrogen on yield depends on the variety.

The analysis shows that variety, nitrogen levels, and their interaction significantly influence yield, with additional variation attributed to tree blocks.

c) Perform a post-hoc test for the interaction effect (variety × nitrogen) and draw a bar diagram with lettering.

 # Code

```
Post.Hoc.Test<-with(data,HSD.test(YIELD,vari.fact:nitro.fact,DFerror = 28,MSerror = 0.044))

Mean.matrix<-Post.Hoc.Test$means

Mean.matrix<-Mean.matrix[order(Mean.matrix$YIELD,decreasing = TRUE),]
Mu_Tret<-Mean.matrix$YIELD
SE_Treat<-Mean.matrix$std/sqrt(Mean.matrix$YIELD)

Bar.Plot <- barplot2(Mu_Tret, names.arg = rownames(Mean.matrix),
                     xlab = "Treatment Combinations",
                     ylab = "Mean Yield", plot.ci = TRUE,
                     ci.l = Mu_Tret - SE_Treat, ci.u = Mu_Tret + SE_Treat,
                     col = "green", las = 2)

letters <- c("a", "ab", "ab", "bc", "bc", "bc", "c", "cd", "de",
             "e", "e", "e", "f", "f", "f")

text(x = Bar.Plot, y = Mu_Tret + SE_Treat + 0.1, labels = letters, cex = 0.8)
```

#RESULT:

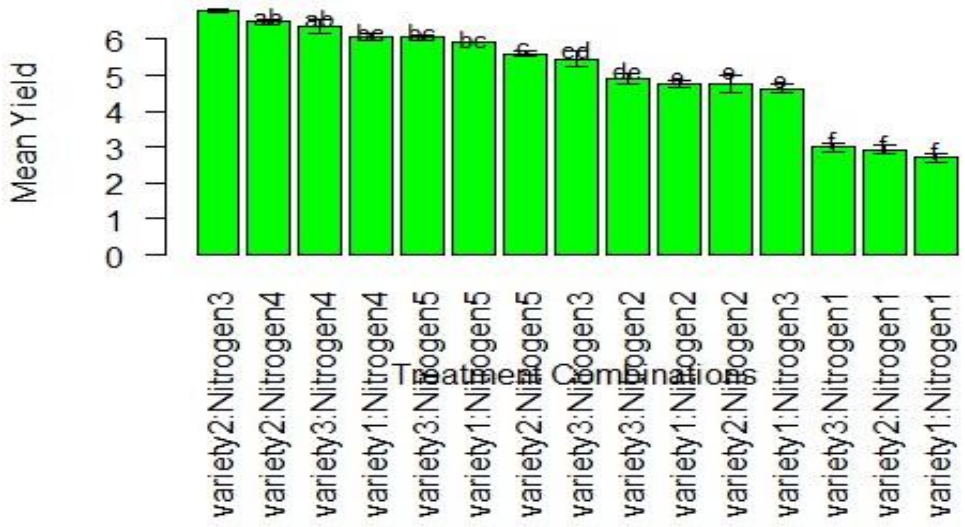
$statistics

MSerror	Df	Mean	CV	MSD
0.044	28	5.094	4.11782	0.6348227

$groups

	YIELD	groups
variety2:Nitrogen3	6.806667	a
variety2:Nitrogen4	6.490000	ab
variety3:Nitrogen4	6.346667	ab
variety1:Nitrogen4	6.070000	bc
variety3:Nitrogen5	6.056667	bc
variety1:Nitrogen5	5.923333	bc
variety2:Nitrogen5	5.596667	c
variety3:Nitrogen3	5.443333	cd
variety3:Nitrogen2	4.910000	de

variety1:Nitrogen2 4.760000 e
variety2:Nitrogen2 4.743333 e
variety1:Nitrogen3 4.636667 e
variety3:Nitrogen1 2.993333 f
variety2:Nitrogen1 2.936667 f
variety1:Nitrogen1 2.696667 f



Solution 02:

a). Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a statistical technique used to simplify complex datasets by reducing their dimensions while retaining most of the original information. It transforms correlated variables into a smaller number of uncorrelated variables called principal components, which capture the maximum variance in the data.

Key Points:

1. **Dimensionality Reduction**: Makes large datasets easier to analyze and visualize.
2. **Variance Focus**: The first few components capture the most important patterns in the data.
3. **Applications**: Used for pattern recognition, data visualization, feature selection, and noise reduction.

b). The main purposes of Principal Component Analysis (PCA) in Biotechnology and Genetic Engineering (my study area):

In Biotechnology and Genetic Engineering, **PCA** (Principal Component Analysis) is used primarily as a statistical tool for data analysis and dimensionality reduction. Specifically, PCA is employed for:

1. **Gene Expression Analysis**: PCA is often used to analyze gene expression data from experiments like microarrays or RNA sequencing. By reducing the dimensionality of the large gene expression datasets, PCA can help identify patterns or groups of genes that behave similarly across different conditions or samples, facilitating the understanding of gene regulatory networks and the identification of biomarkers.
2. **Genomic Data Interpretation**: In genomic studies, PCA helps to visualize and understand large-scale datasets such as single nucleotide polymorphisms (SNPs) or whole-genome sequences. It aids in identifying underlying patterns in genetic variation among individuals or populations.
3. **Clustering and Classification**: PCA can be used to preprocess and reduce data dimensions before applying clustering algorithms (like k-means) or classification models (such as SVM or decision trees). This simplification helps improve the efficiency and accuracy of these models.
4. **Quality Control**: PCA is useful for identifying outliers or errors in experimental data, ensuring that the data used for further analysis is consistent and reliable. For example, it can highlight samples that deviate significantly from others in a study.
5. **Metabolomics and Proteomics**: In metabolomics and proteomics, PCA helps to visualize complex data from large-scale biochemical analyses, revealing patterns in metabolite or protein expression that could indicate disease states or biological processes.

Overall, PCA is a powerful tool for extracting meaningful patterns and reducing the complexity of large datasets in biotechnology and genetic engineering.

c). Computation of the the eigenvalue and eigenvector using the iris data based on R programming is given below-

Code

```
# Load the data

iris_data <- read.csv("iris_Data.csv")

# Extract numerical columns (exclude the species column)

numeric_data <- iris_data[, 1:4]

# Compute the covariance matrix
```

```
cov_matrix <- cov(numeric_data)
# Compute eigenvalues and eigenvectors
eigen_results <- eigen(cov_matrix)

# Display the eigenvalues
cat("Eigenvalues:\n")
print(eigen_results$values)

# Display the eigenvectors
cat("\nEigenvectors:\n")
print(eigen_results$vectors)
```

Result:

Eigenvalues:

```
[1] 4.22824171 0.24267075 0.07820950 0.02383509
```

Eigenvectors:

```
           [,1]       [,2]       [,3]       [,4]
[1,]  0.36138659 -0.65658877  0.58202985  0.3154872
[2,] -0.08452251 -0.73016143 -0.59791083 -0.3197231
[3,]  0.85667061  0.17337266 -0.07623608 -0.4798390
[4,]  0.35828920  0.07548102 -0.54583143  0.7536574
```

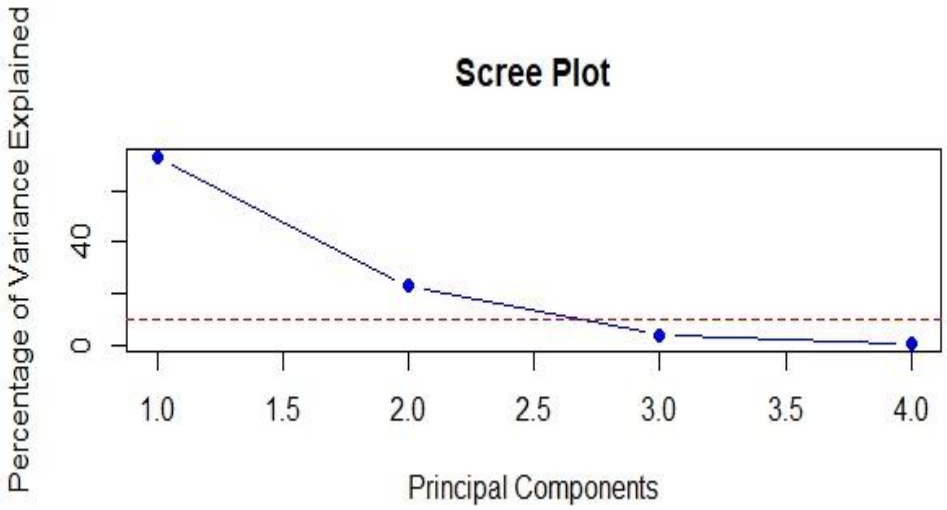
d). Construction of a scree plot and interpretation of how many principle components should be retained to interpret the iris dataset is given below:

Code

```
# Load the data
```

```
iris_data <- read.csv("iris_Data.csv")

# Extract numerical columns (exclude the species column)
numeric_data <- iris_data[, 1:4]
# Perform PCA
pca_result <- prcomp(numeric_data, scale. = TRUE) # Scale the data for standardization

# Compute the proportion of variance explained
explained_variance <- (pca_result$sdev^2) / sum(pca_result$sdev^2) * 100

# Cumulative variance explained
cumulative_variance <- cumsum(explained_variance)

# Create a scree plot
plot(
  explained_variance,
  type = "b",
  xlab = "Principal Components",
  ylab = "Percentage of Variance Explained",
  main = "Scree Plot",
  pch = 19,
  col = "blue"
)
abline(h = 10, col = "red", lty = 2) # Optional: threshold for significance
# Add cumulative variance interpretation (optional)
cat("Explained Variance by Principal Components:\n")
print(explained_variance)
```

```
cat("\nCumulative Variance:\n")
print(cumulative_variance)
```




```
pca_result
Standard deviations (1, .., p=4):
[1]   1.7083611   0.9560494   0.3830886   0.1439265
Rotation (n x k) = (4 x 4):
                    PC1         PC2         PC3         PC4
Sepal.Length   0.5210659 -0.37741762   0.7195664   0.2612863
Sepal.Width   -0.2693474 -0.92329566 -0.2443818 -0.1235096
Petal.Length   0.5804131 -0.02449161 -0.1421264 -0.8014492
Petal.Width   0.5648565 -0.06694199 -0.6342727   0.5235971

Explained Variance by Principal Components:
[1] 72.9624454   22.8507618   3.6689219   0.5178709
Cumulative Variance:
[1]   72.96245   95.81321   99.48213   100.00000
```

Interpretation:

Scree Plot Insight:

In the scree plot, observed a sharp drop in variance explained from PC1 to PC2, and then the curve flattens after PC2. This suggests that **two principal components** would be adequate to interpret the dataset.

It can be choosen to retain **two components** for dimensionality reduction, as this will capture most of the variance without losing much information.

The scree plot shows the **percentage of variance explained** by each principal component (PC):

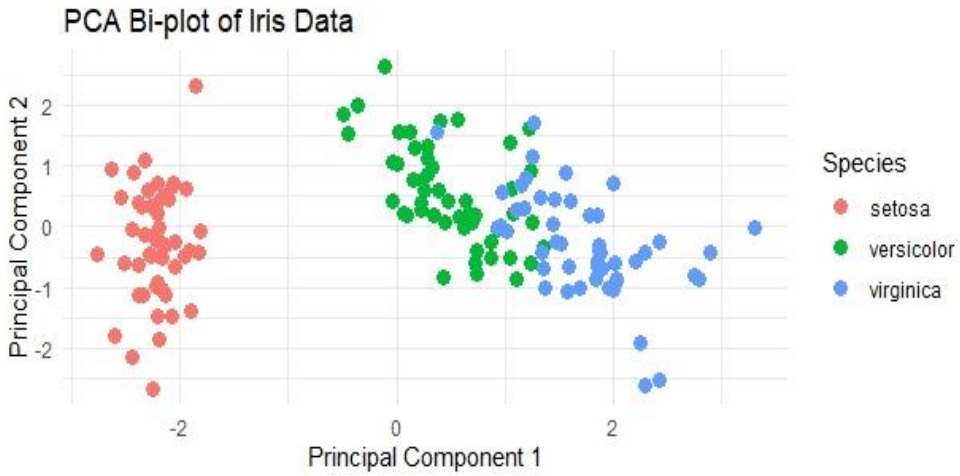
1. **PC1** (first component):
    - Explains the largest variance (around 72.96% as per your data).
    - Represents the most significant pattern in the dataset.
2. **PC2** (second component):
    - Adds a significant amount of variance (around 22.85%, bringing the cumulative variance to 95.81%).
    - Together, PC1 and PC2 capture the majority of the information (approximately 96%).
3. **PC3 and PC4**:
    - Contribute very little additional variance (3.67% and 0.52%, respectively).
    - These components are not significant for explaining the variability in the data.

Retain PC1 and PC2: These two components explain around **96% of the total variance**, which is sufficient to summarize the dataset effectively.
Discard PC3 and PC4: These components add minimal new information and can be ignored in most analyses.

e**). Construction a bi-plot for the iris data based on R programming and interpretion of the the results is given below:**

```
# Load the iris dataset
data(iris)

# Perform PCA on the numerical columns of the iris dataset (excluding the Species column)
pca_result <- prcomp(iris[, 1:4], center = TRUE, scale. = TRUE)

# Plot the bi-plot
biplot(pca_result, main = "Bi-plot of Iris Data")

# Optionally, you can customize the plot with different colors for each species
library(ggplot2)
pca_data <- data.frame(pca_result$x, Species = iris$Species)

# Plot with ggplot2 for better customization
ggplot(pca_data, aes(PC1, PC2, color = Species)) +
  geom_point(size = 3) +
  labs(title = "PCA Bi-plot of Iris Data", x = "Principal Component 1", y = "Principal Component 2") +
  theme_minimal()
```

Ans:



Interpretion:

- **Species Labels**: Each point is labeled with its species (setosa, versicolor, or virginica), making it easy to see how the species are distributed along the principal components.
- **Cluster Separation**: To observe clear separation of points between species (e.g., setosa may cluster in one part of the plot while versicolor and virginica cluster in other parts), this suggests that the principal components (PC1 and PC2) capture the variation that distinguishes these species.
- **Principal Components**: The arrows in the bi-plot represent the loadings of the original variables (sepal length, sepal width, petal length, and petal width) on the principal components.