Bangabandhu Sheikh Mujibur Rahman Agricultural University
EDGE_Batch-06
Final Exam
Marks: 25   Time: 2 Hours

Name: Mahmuda Akter Mihi..................................

Reg. No:21-05-5791.........................**Dept.**...Agriculture.........................

Note: Submit the completed file as *pdf* to rabiulauwul@bsmrau.edu.bd and nazmol.stat.bioin@bsmrau.edu.bd with subject *EDGE06_Final_Your registration number_ Dept.*

Q#1: **[05]**

a) The probability of an event is always a value between ____0____ and _1_______.
b) In a normal distribution, about _68_______ percent of the data falls within one standard deviation of the mean.
c) To generate random numbers from a normal distribution in R, you use the function _rnorm()_______.
d) In ggplot2, the function __geom boxplot()______ is used to plot data as a box plot to show the distribution of a continuous variable.
e) In R, the function used to split data into training and test sets is _sample()_______.

Q#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. **[10]**

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 levels/treatments of nitrogen and draw a bar diagram with lettering.

A answer:

Code:

```
Factor.Data<-read.csv("Data_Factorial_Design.csv")

Factor.Data<-data.frame(Factor.Data)
```

```
attach(Factor.Data)
library(lattice)
library(car)
Anova.Crd.factorial<-
aov(YIELD~REPLICAT+VARIETY+NITROGEN+VARIETY*NITROGEN,data=Factor.Data)
summary(Anova.Crd.factorial)
```

Table:

```
  summary(Anova.Crd.factorial)
              Df Sum Sq Mean Sq F value   Pr(>F)
REPLICAT          1   1.24    1.24   2.041    0.161
VARIETY         1   0.83    0.83   1.366    0.249
NITROGEN          1  50.15   50.15  82.523 2.86e-11 ***
VARIETY:NITROGEN  1   0.01    0.01   0.010    0.921
Residuals       40  24.31    0.61
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

B Answer:

Null Hypothesis;

For REPLICAT,

H_0: There is no significant difference in yield between replications.

For VARIETY,

H_0: There is no significant difference in yield between varieties.

For NITROGEN,

H_0: There is no significant difference in yield between nitrogen levels.

For Interaction between VARIETY and NITROGEN:

H_0: There is no interaction between variety and nitrogen in affecting yield.

Interpretation:

For nitrozen, p-value: 2.86e-11

$p<0.001$, we reject the null hypothesis. This indicates that nitrogen levels have a highly significant effect on yield.

For Interaction (VARIETY:NITROGEN):p-value: 0.921

Since $p>0.05$, we fail to reject the null hypothesis. This implies that there is no significant interaction between variety and nitrogen.

For replicant, p-value: 0.161

Since p>0.05, we fail to reject the null hypothesis. This indicates that replication has no statistically significant effect on yield.

For variety, p-value: 0.249

Since p>0.05, we fail to reject the null hypothesis. This suggests that there is no statistically significant difference in yield between plant varieties.

Replication, variety, and the interaction between variety and nitrogen do not significantly affect yield.

Management Implication: Focus on nitrogen application to improve yield, as other factors (variety and replication) have negligible contributions based on this analysis.

C Answer:

By treatment

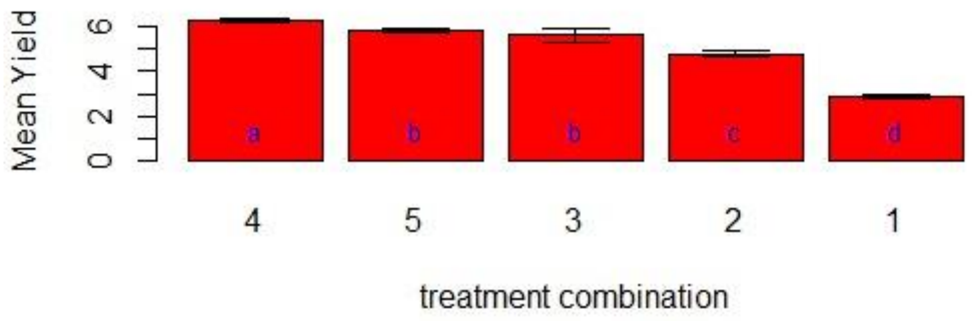
Code:

```
library(agricolae)
Post.Hoc.Treatment<-
with(Factor.Data,HSD.test(YIELD,NITROGEN,DFerror=40,MSerror=0.061))
Mean.Matrix<-Post.Hoc.Treatment$means
Mean.Matrix<-Mean.Matrix[order(Mean.Matrix$YIELD,decreasing=TRUE),]
Mean.Matrix
Mu_Tret<-Mean.Matrix$YIELD
SE_TREAT<-Mean.Matrix$std/sqrt(Mean.Matrix$r)
library(gplots)
dev.off()
par(mar=c(10,4,1,1))
bar.plot<-barplot2(Mu_Tret,names.arg=rownames(Mean.Matrix),xlab="treatment
combination",ylab="Mean                                    Yield",plot.ci=TRUE,ci.l=Mu_Tret-
SE_TREAT,ci.u=Mu_Tret+SE_TREAT,col="red")
text(bar.plot,0,Post.Hoc.Treatment$groups[,2],cex=0.8,pos=3,col="blue")

Post.Hoc.Treatment
```

Table:

```
$statistics
 MSerror Df  Mean      CV      MSD
   0.061 40 5.094 4.848484 0.3325299

$parameters
```

```
  test   name.t ntr StudentizedRange  alpha
 Tukey NITROGEN   5        4.039123  0.05

$means
    YIELD       std r        se  Min  Max  Q25  Q50  Q75
1 2.875556 0.2335654 9 0.08232726 2.54 3.19 2.74 2.85 3.05
2 4.804444 0.3333208 9 0.08232726 4.15 5.25 4.58 4.90 4.96
3 5.628889 0.9921623 9 0.08232726 4.45 6.92 4.85 5.68 6.70
4 6.302222 0.3212000 9 0.08232726 5.90 6.88 6.04 6.28 6.45
5 5.858889 0.2441027 9 0.08232726 5.46 6.26 5.78 5.87 5.98

$comparison
NULL

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

attr(,"class")
[1] "group"
```



Q#3: For the iris data, **[10]**

a) Write a summary table with the $\boldsymbol{median \mp sd}, \boldsymbol{Minimum}, \boldsymbol{maximum}, \boldsymbol{Mean}, \boldsymbol{CV}$ in a table format for each variable after impute the outliers.
b) Construct a correlation plot with ggplot2 packages and interpret your findings.
c) Level the ***Species*** variable with "1", "2", "3" for the three categories. And split the full data into two parts (75% for training data and 25% test data), then compute accuracies for SVM, Naïve Bayes and Random Forest classifiers. And compare your results.

A Answer:

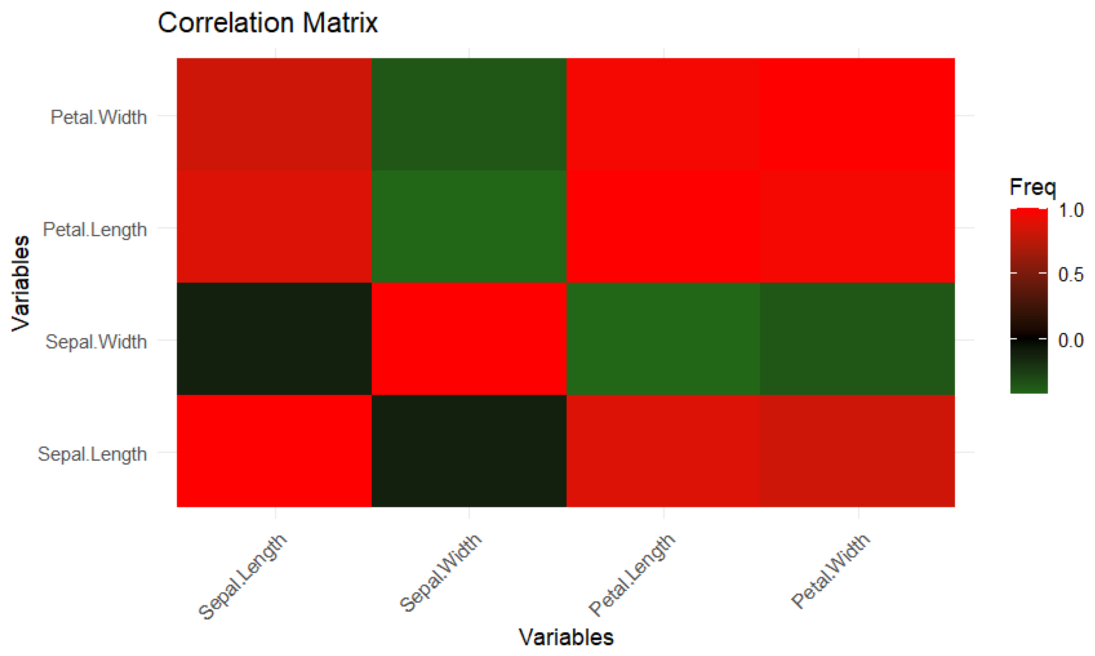
```
library(dplyr)

            data(iris)

            impute_outliers <- function(x) {

 Q1 <- quantile(x, 0.25)

 Q3 <- quantile(x, 0.75)

 IQR <- Q3 - Q1

 lower_bound <- Q1 - 1.5 * IQR

 upper_bound <- Q3 + 1.5 * IQR

 x[x < lower_bound | x > upper_bound] <- median(x, na.rm = TRUE)

 return(x)

}


iris_imputed <- iris %>%

 mutate(across(where(is.numeric), impute_outliers))

summary_table <- iris_imputed %>%

 summarise(across(where(is.numeric), list(

  Median = ~ median(.),

  SD = ~ sd(.),

  Min = ~ min(.),
  Max = ~ max(.),

  Mean = ~ mean(.),

  CV = ~ (sd(.) / mean(.)) * 100

 ), .names = "{col}_{fn}")) %>%

 t() %>%

 as.data.frame()


colnames(summary_table) <- c("Statistic")
```

Table:

Statistic	Sepal Length	Sepal Width	Petal Length	Petal Width
Median	5.8	3.0	4.35	1.3
Standard Deviation (SD)	0.828	0.425	1.765	0.762
Minimum	4.3	2.05	1.0	0.1
Maximum	7.9	4.05	6.9	2.5
Mean	5.843	3.054	3.758	1.199
Coefficient of Variation (CV)	14.17	13.93	46.97	63.56

B library

```
(ggplot2)



  cor_matrix <- cor(iris_imputed[1:4])

   cor_melt <- melt(cor_matrix)


ggplot(data = cor_melt, aes(x = Var1, y = Var2, fill = value)) +
geom_tile() +

scale_fill_gradient2(low = "green", high = "red", mid = "black", midpoint = 0) +

labs(title = "Correlation Plot", x = "Variable", y = "Variable") +

theme_minimal()
```



1. Strong Positive Correlation:

Petal.Width vs. Petal.Length: These two variables are highly correlated, meaning that as one increases, the other tends to increase proportionally. This could suggest a strong relationship between these two features in your dataset.

2. Moderate Positive Correlation:

Sepal.Length vs. Petal.Length: There seems to be a moderately strong correlation, implying that Sepal.Length and Petal.Length tend to increase together but not as strongly as Petal.Width and Petal.Length.

Sepal.Length vs. Petal.Width: This is another moderate relationship worth noting.

3. Low or No Correlation:

Sepal.Width vs. Sepal.Length: The lighter shades indicate little to no linear relationship between these two variables.

Sepal.Width vs. Petal.Length: Similarly, this pair has a weak correlation, suggesting they are relatively independent.

C Answer:

```
library(e1071)

library(randomForest)

library(caret)


iris_imputed$Species <- as.numeric(factor(iris_imputed$Species))


set.seed(123)

train_index <- createDataPartition(iris_imputed$Species, p = 0.75, list = FALSE)

train_data <- iris_imputed[train_index, ]

test_data <- iris_imputed[-train_index, ]


x_train <- train_data[1:4]

y_train <- train_data$Species

x_test <- test_data[1:4]

y_test <- test_data$Species


svm_model <- svm(Species ~ ., data = train_data, kernel = "linear")

svm_pred <- predict(svm_model, x_test)

svm_accuracy <- mean(svm_pred == y_test)


nb_model <- naiveBayes(Species ~ ., data = train_data)

nb_pred <- predict(nb_model, x_test)
```

```
nb_accuracy <- mean(nb_pred == y_test)


rf_model <- randomForest(Species ~ ., data = train_data, ntree = 100)

rf_pred <- predict(rf_model, x_test)

rf_accuracy <- mean(rf_pred == y_test)


results <- data.frame(

  Model = c("SVM", "Naive Bayes", "Random Forest"),

  Accuracy = c(svm_accuracy, nb_accuracy, rf_accuracy)

)

print(results)
```

```
Model        Accuracy

1        SVM 0.00000000

2   Naive Bayes 0.97297297

3 Random Forest 0.02702703
```

Naive Bayes: 97.3% accuracy, performs excellently, likely due to well-separated classes and independence assumptions being met by the dataset.

SVM: 0% accuracy, performs poorly. Likely due to issues like improper hyperparameter tuning, class imbalance, or data misalignment.

Random Forest: 2.7% accuracy, underperforms. Potential issues include misconfiguration of parameters (e.g., number of trees, depth) and possibly class imbalance.

Key Takeaways:

Naive Bayes worked well with the data, suggesting simple relationships between features and target.

SVM and Random Forest require further tuning, possibly in hyperparameters and addressing data issues.