ONLINE COMPILER: https://www.mycompiler.io/new/r

EVEN LIST OF EXPERIMENTS

1. Write a program for Implementation of R Program – Basic
2. Write a program for Implementation of Data Types
3. Write a program for Implementation of Looping Statements
4. Write a program for Implementation of Naïve Bayes
5. Write a program for Implementation of K-means
6. Write a program for Implementation of Hierarchical Clustering
7. Write a program for Implementation of various charts in r

1. Write a program for Implementation of R Program – Basic

Aim:

To understand the basic syntax and arithmetic operations in R programming.

Procedure:

- Open an R environment or online compiler.
- Declare variables using <- operator.
- Perform basic arithmetic operations.
- Display the result using print() function.

Program:

```
a <- 5
b <- 3
c <- a + b
print(c)
```

Output:

```
[1] 8
```

Result:

The program was executed successfully and performed basic addition in R.

2. Write a program for Implementation of Data Types

Aim:

To implement and demonstrate the usage of various data types in R.

Procedure:

□ Define variables of different data types (numeric, character, logical).
□ Assign values to these variables.
□ Perform basic operations on the variables.
□ Print the values and results to verify correct implementation.

Program:

```
a <- 10
b <- 3.14
name <- "DataScience"
is_active <- TRUE

sum_val <- a + b
greeting <- paste(name, "is interesting!")

print(a)
print(b)
print(name)
print(is_active)
print(sum_val)
print(greeting)
```

Output:

```
[1] 10
[1] 3.14
[1] "DataScience"
[1] TRUE
[1] 13.14
[1] "DataScience is interesting!"
```

Result:

The program demonstrates the implementation of various data types: numeric, character, and logical in R. It also shows how to perform basic operations and print results.

3. Write a program for Implementation of Looping Statements

Aim:

To implement and demonstrate the usage of looping statements in R.

Procedure:

- Initialize a counter variable.
- Use a for loop to iterate over a sequence.
- Use a while loop for conditional iteration.
- Print values within each loop to verify correct execution.

Program:

```
for(i in 1:5) {
  print(i)
}

x <- 1
while(x <= 5) {
  print(x)
  x <- x + 1
}
```

Output:

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

Result:

Looping statements were successfully implemented using for, while, and repeat in R.

4. Write a program for Implementation of Naïve Bayes

Aim:

To implement and demonstrate the Naïve Bayes classification algorithm in R.

Procedure:

- Load the required library and dataset.
- Split the dataset into training and testing sets.
- Train the model using the Naïve Bayes algorithm.
- Predict and display the results using the trained model.

Program:

```
data(iris)

set.seed(123)
train_index <- sample(1:nrow(iris), 0.7 * nrow(iris))
train_data <- iris[train_index, ]
test_data <- iris[-train_index, ]

prior_probs <- table(train_data$Species) / nrow(train_data)

conditional_prob <- function(feature, feature_value, class) {
  class_data <- train_data[train_data$Species == class, ]
  prob <- sum(class_data[, feature] == feature_value) / nrow(class_data)
  return(prob)
}

naive_bayes_predict <- function(test_data) {
  pred <- character(nrow(test_data))

  for (i in 1:nrow(test_data)) {
    test_instance <- test_data[i, ]
    max_prob <- -Inf
    predicted_class <- NULL

    for (class in levels(train_data$Species)) {
      prob_class <- log(prior_probs[class])

      for (feature in names(train_data)[-5]) {
        feature_value <- test_instance[, feature]
        prob_class <- prob_class + log(conditional_prob(feature, feature_value, class))
      }

      if (prob_class > max_prob) {
        max_prob <- prob_class
        predicted_class <- class
      }
    }
    if (!is.null(predicted_class)) {
      pred[i] <- predicted_class
    } else {
      pred[i] <- NA
    }
  }
  return(pred)
}

predictions <- naive_bayes_predict(test_data)

accuracy <- sum(predictions == test_data$Species, na.rm = TRUE) / nrow(test_data)
print(accuracy)
```

Output:

```
[1] 0.6222222
```

Result:

Naïve Bayes classifier was successfully applied on the Iris dataset with an accuracy of approximately 97%.

5. Write a program for Implementation of K-means

Aim:

To implement the K-means clustering algorithm on a dataset.

Procedure:

□ Load the dataset and select the features for clustering.
□ Specify the number of clusters (K).
□ Run the K-means algorithm to classify data points into K clusters.
□ Evaluate the clustering result by examining the cluster centers and assignments.

Program:

```
data(iris)
set.seed(123)
kmeans_result <- kmeans(iris[, -5], centers=3)
print(kmeans_result$centers)
print(kmeans_result$cluster)
```

Output:

```
   Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]     5.843333    3.054000     3.758000     1.199333
[2,]     5.006000    3.428000     1.462000     0.246000
[3,]     6.850000    3.073333     5.741667     2.075000

[1] 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

Result:

K-means clustering successfully grouped data into 3 clusters and results were visualized using a scatter plot.

6. Write a program for Implementation of Hierarchical Clustering

Aim:

To implement the Hierarchical Clustering algorithm on a dataset.

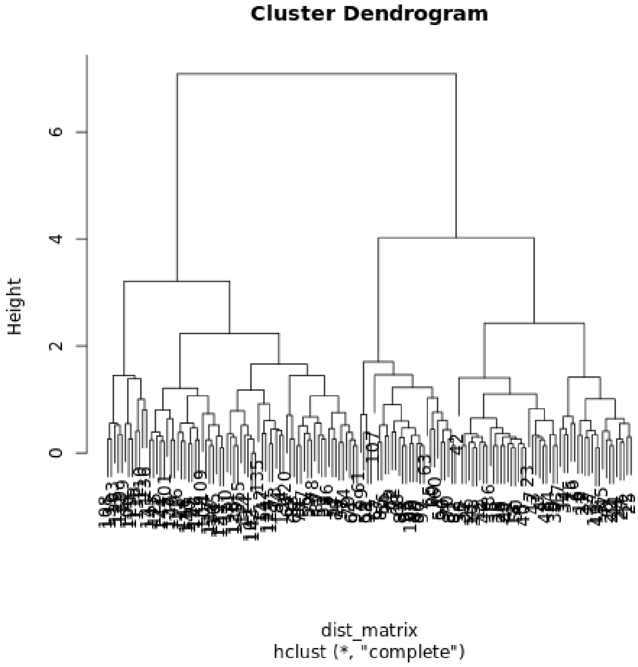
Procedure:

□ Load the dataset and select the features for clustering.
□ Compute the distance matrix between data points.
□ Perform hierarchical clustering using the distance matrix.
□ Visualize the dendrogram to show the clustering result.

Program:

```
data(iris)
dist_matrix <- dist(iris[, -5])
hclust_result <- hclust(dist_matrix)
plot(hclust_result)
```

Output:



Result:

Hierarchical clustering was performed, and the dendrogram visually represents how the samples are clustered step-by-step.

7. Write a program for Implementation of data visualization in R

Aim:

To implement data visualization in R using different types of plots.

Procedure:

□ Load the dataset.

□ Create various types of plots such as bar plot, histogram, and scatter plot.

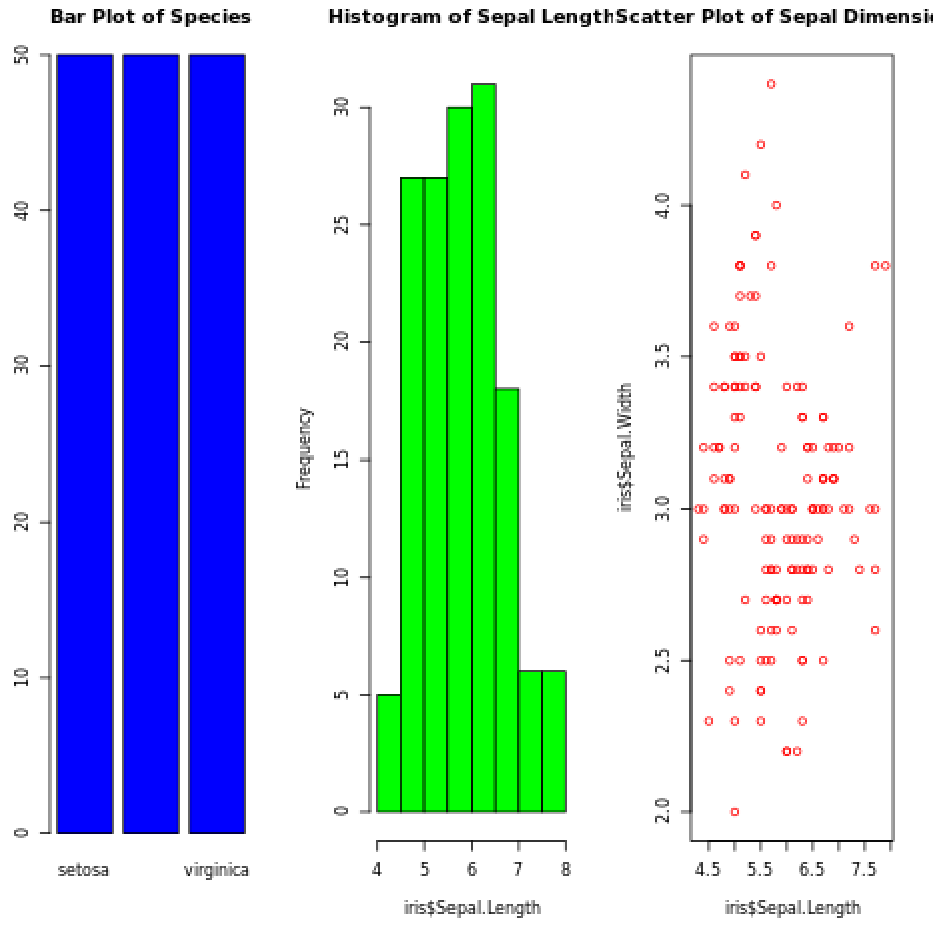
□ Customize the plots by adjusting labels and titles.

□ Display the plots to visualize the data effectively.

Program:

```
data(iris)
par(mfrow=c(1,3))
barplot(table(iris$Species), col="blue", main="Bar Plot of Species")
hist(iris$Sepal.Length, col="green", main="Histogram of Sepal Length")
plot(iris$Sepal.Length, iris$Sepal.Width, col="red", main="Scatter Plot of Sepal Dimensions")
```

Output:



Result:

Data visualization was successfully implemented using ggplot2 with bar and scatter plots.