OPERATING SYSTEMS LAB (2024-2025)

II. B. Tech I Semester

G. Mrunalini
Asst. Professor

LEAD EXPERIMENTS

1. write a C program to stimulate the CPU Scheduling algorithm SRTF.

2. Write a C program to simulate the concept of Dining Philosopher problem

1.write a C program to stimulate the CPU Scheduling algorithm SRTF.

```
#include <stdio.h>

#define MAX_PROCESSES 100

struct Process {
    int id;
    int arrival_time;
    int burst_time;
    int remaining_time;
    int completion_time;
    int waiting_time;
    int turnaround_time;
};

int main() {
    struct Process p[MAX_PROCESSES];
    int n, i, time = 0, completed = 0, min_index;
    float total_waiting_time = 0, total_turnaround_time = 0;

    printf("Enter number of processes: ");
    scanf("%d", &n);

    // Input arrival and burst time
    for (i = 0; i < n; i++) {
        p[i].id = i + 1;
        printf("Enter Arrival Time and Burst Time for Process P%d: ", p[i].id);
        scanf("%d %d", &p[i].arrival_time, &p[i].burst_time);
        p[i].remaining_time = p[i].burst_time;
    }
```

```
    // SRTF Scheduling
    int prev = -1;
    while (completed != n) {
        min_index = -1;
        int min_time = 9999;

        for (i = 0; i < n; i++) {
            if (p[i].arrival_time <= time && p[i].remaining_time > 0 &&
p[i].remaining_time < min_time) {
                min_time = p[i].remaining_time;
                min_index = i;
            }
        }

        if (min_index == -1) {
            time++;
            continue;
        }

        // Execute 1 unit time
        p[min_index].remaining_time--;
        time++;

        // If process is completed
        if (p[min_index].remaining_time == 0) {
            completed++;
            p[min_index].completion_time = time;
            p[min_index].turnaround_time = p[min_index].completion_time -
p[min_index].arrival_time;
            p[min_index].waiting_time = p[min_index].turnaround_time -
p[min_index].burst_time;

            total_waiting_time += p[min_index].waiting_time;
            total_turnaround_time += p[min_index].turnaround_time;
```

```
        }
    }

    // Output
    printf("\nProcess\tAT\tBT\tCT\tTAT\tWT\n");
    for (i = 0; i < n; i++) {
        printf("P%d\t%d\t%d\t%d\t%d\t%d\n", p[i].id, p[i].arrival_time, p[i].burst_time,
p[i].completion_time,
                p[i].turnaround_time, p[i].waiting_time);
    }

    printf("\nAverage Turnaround Time = %.2f", total_turnaround_time / n);
    printf("\nAverage Waiting Time = %.2f\n", total_waiting_time / n);

    return 0;
}
```

 Sample Output

pgsql

Copy

Edit

```
Enter number of processes: 3
Enter Arrival Time and Burst Time for Process P1: 0 8
Enter Arrival Time and Burst Time for Process P2: 1 4
Enter Arrival Time and Burst Time for Process P3: 2 9

Process AT  BT  CT  TAT WT
P1      0   8   17  17  9
P2      1   4   5   4   0
P3      2   9   26  24  15

Average Turnaround Time = 15.00
Average Waiting Time = 8.00
```

2.Write a C program to simulate the concept of Dining Philosopher problem

```
include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

#define N 5 // Number of philosophers

#define THINKING 0
#define HUNGRY 1
#define EATING 2

int state[N];
int phil_ids[N] = {0, 1, 2, 3, 4};

sem_t mutex;
sem_t S[N];

void test(int i) {
    if (state[i] == HUNGRY &&
        state[(i + 4) % N] != EATING &&
        state[(i + 1) % N] != EATING) {

        state[i] = EATING;
        sleep(1); // simulate time to eat
        printf("Philosopher %d takes forks %d and %d\n", i + 1, (i + 4) % N + 1, (i + 1) %
N + 1);
        printf("Philosopher %d is EATING\n", i + 1);
        sem_post(&S[i]);
    }
```

```
}

void take_forks(int i) {
    sem_wait(&mutex); // enter critical section
    state[i] = HUNGRY;
    printf("Philosopher %d is HUNGRY\n", i + 1);
    test(i);
    sem_post(&mutex); // exit critical section
    sem_wait(&S[i]);  // wait if not able to eat
}
void put_forks(int i) {
    sem_wait(&mutex); // enter critical section
    state[i] = THINKING;
    printf("Philosopher %d puts down forks %d and %d\n", i + 1, (i + 4) % N + 1, (i +
1) % N + 1);
    printf("Philosopher %d is THINKING\n", i + 1);
    test((i + 4) % N); // check left neighbor
    test((i + 1) % N); // check right neighbor
    sem_post(&mutex); // exit critical section
}

void* philosopher(void* num) {
    int i = *(int*)num;
    while (1) {
        sleep(1);          // thinking
        take_forks(i);     // pick up forks
        sleep(2);          // eating
        put_forks(i);      // put down forks
    }
}
int main() {
    int i;
    pthread_t thread_id[N];
```

```
    sem_init(&mutex, 0, 1);
    for (i = 0; i < N; i++)
        sem_init(&S[i], 0, 0);

    for (i = 0; i < N; i++)
        pthread_create(&thread_id[i], NULL, philosopher, &phil_ids[i]);

    for (i = 0; i < N; i++)
        pthread_join(thread_id[i], NULL);

    return 0;
}
```

Sample Output

After compiling and running the program, you might see output similar to this (will vary on each run due to thread scheduling):

csharp

Copy

Edit

Philosopher 1 is HUNGRY

Philosopher 1 takes forks 5 and 2

Philosopher 1 is EATING

Philosopher 2 is HUNGRY

Philosopher 3 is HUNGRY

Philosopher 4 is HUNGRY

Philosopher 5 is HUNGRY

Philosopher 1 puts down forks 5 and 2

Philosopher 1 is THINKING

Philosopher 2 takes forks 1 and 3

Philosopher 2 is EATING

Philosopher 2 puts down forks 1 and 3

Philosopher 2 is THINKING

Philosopher 3 takes forks 2 and 4

Philosopher 3 is EATING

Java Programming Lab

II. B. Tech I. Semester

Radhe Shyam Panda
Asst. Professor

2024-2025

LEAD EXPERIMENTS

1) Java program to change the host name to its specific IP Address.

2) Java Program to validate an email address.

3) Java program to print date in different format.

1) Java program to change the host name to its specific IP Address.

Aim: To implement a java to change the host name to its specific IP Address.

Program:

```
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.util.Scanner;
public class HostnameToIP {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter hostname (e.g., www.google.com): ");
      String hostname = scanner.nextLine();

      try {
         // Get InetAddress object for the hostname
         InetAddress inetAddress = InetAddress.getByName(hostname);

         // Print the IP address
         System.out.println("IP address of " + hostname + " is: " +
inetAddress.getHostAddress());
      } catch (UnknownHostException e) {
         System.out.println("Hostname could not be resolved: " + e.getMessage());
      }

      scanner.close();
   }
}
```

Result:

Input:

Enter hostname (e.g., www.google.com): www.google.com

Output:

IP address of www.google.com is: 142.250.182.132

2). Java Program to validate an email address.

Aim: To implement a Java Program to validate an email address.

Program:

```
import java.util.Scanner;
import java.util.regex.Pattern;
import java.util.regex.Matcher;
public class EmailValidator {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter an email address: ");
      String email = scanner.nextLine();
      // Define email validation regex
      String emailRegex = "^[A-Za-z0-9+_.-]+@[A-Za-z0-9.-]+$";
      // Create Pattern object
      Pattern pattern = Pattern.compile(emailRegex);

      // Match input with the regex
      Matcher matcher = pattern.matcher(email);

      // Print result
      if (matcher.matches()) {
         System.out.println("Valid email address.");
      } else {
         System.out.println("Invalid email address.");
      }

      scanner.close();
   }
}
```

Result:

Input:

Enter an email address: example.user123@gmail.com

Output:

Valid email address.

Input:

Enter an email address: invalid_email@.com

Output:

Invalid email address.

3). **Java program to print date in different format.**

Aim: To implement a Java Program to print date in different format.

Program:

```
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;

public class DateFormatsExample {
   public static void main(String[] args) {
      // Get current date
      LocalDate currentDate = LocalDate.now();

      // Define different formatters
      DateTimeFormatter format1 = DateTimeFormatter.ofPattern("dd/MM/yyyy");
      DateTimeFormatter format2 = DateTimeFormatter.ofPattern("MM-dd-yyyy");
      DateTimeFormatter format3 = DateTimeFormatter.ofPattern("yyyy.MM.dd");
      DateTimeFormatter format4 = DateTimeFormatter.ofPattern("E, MMM dd yyyy");
      DateTimeFormatter format5 = DateTimeFormatter.ofPattern("EEEE, dd MMMM yyyy");

      // Print in different formats
      System.out.println("Default Format: " + currentDate);
      System.out.println("Format dd/MM/yyyy: " + currentDate.format(format1));
      System.out.println("Format MM-dd-yyyy: " + currentDate.format(format2));
      System.out.println("Format yyyy.MM.dd: " + currentDate.format(format3));
      System.out.println("Format E, MMM dd yyyy: " + currentDate.format(format4));
      System.out.println("Format EEEE, dd MMMM yyyy: " + currentDate.format(format5));
   }
}
```

Result:

Output (Assuming Today is June 23, 2025):

```
Default Format: 2025-06-23
Format dd/MM/yyyy: 23/06/2025
Format MM-dd-yyyy: 06-23-2025
Format yyyy.MM.dd: 2025.06.23
Format E, MMM dd yyyy: Mon, Jun 23 2025
Format EEEE, dd MMMM yyyy: Monday, 23 June 2025
```

Deep Learning Lab

IV. B. Tech I Semester

M. Sabitha
Asst. Professor

2024-2025

LEAD EXPERIMENTS

1) Build a deep neural network model start with linear regression using a single variable.

2) Build a Feed Forward neural network for prediction of logic gates.

1) Build a deep neural network model start with linear regression using a single variable.

Aim: To Build a deep neural network model start with linear regression using a single variable.

Definition: Linear Regression with One Variable

A simple model that finds the best-fitting straight line between a single input (x) and output (y).

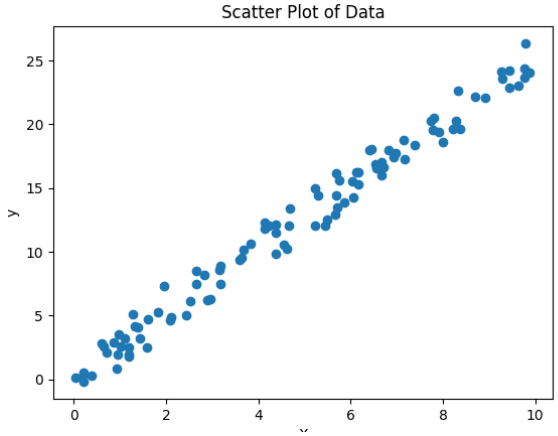
Deep Neural Network (DNN)

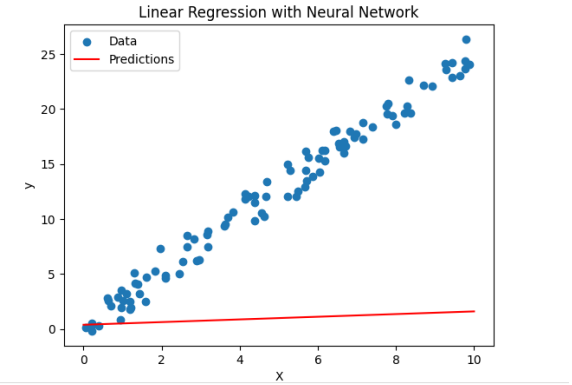
An advanced model made by stacking layers of neurons. Starting from linear regression, a DNN adds hidden layers to learn complex, non-linear patterns from the same input.

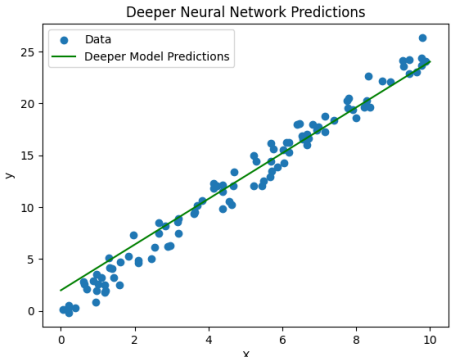
Source code:

```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
# Generate some synthetic data
np.random.seed(0)
X = np.random.rand(100, 1) * 10  # 100 samples of a single feature
y = 2.5 * X + np.random.randn(100, 1)  # Linear relationship with noise
plt.scatter(X, y)
plt.title("Scatter Plot of Data")
plt.xlabel("X")
plt.ylabel("y")
plt.show()
# Create a neural network model
model = keras.Sequential([layers.Dense(1, input_shape=(1,), activation='linear')  # Linear
regression layer])
# Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')
# Train the model
model.fit(X, y, epochs=100, verbose=1)
# Make predictions
X_test = np.array([[0], [5], [10]])
predictions = model.predict(X_test)
# Visualize predictions
plt.scatter(X, y, label='Data')
plt.plot(X_test, predictions, color='red', label='Predictions')
plt.title("Linear Regression with Neural Network")
```

```
plt.xlabel("X")

plt.ylabel("y")
plt.legend()
plt.show()
# Create a deeper neural network model
deeper_model = keras.Sequential([
    layers.Dense(10, activation='relu', input_shape=(1,)),  # Hidden layer
    layers.Dense(10, activation='relu'),                    # Hidden layer
    layers.Dense(1, activation='linear')                    # Output layer
])

# Compile the deeper model
deeper_model.compile(optimizer='adam', loss='mean_squared_error')

# Train the deeper model
deeper_model.fit(X, y, epochs=100, verbose=1)

# Make predictions with the deeper model
deeper_predictions = deeper_model.predict(X_test)

# Visualize predictions from the deeper model
plt.scatter(X, y, label='Data')
plt.plot(X_test, deeper_predictions, color='green', label='Deeper Model Predictions')
plt.title("Deeper Neural Network Predictions")
plt.xlabel("X")
plt.ylabel("y")
plt.legend()
plt.show()
```







2)Build a Feed Forward neural network for prediction of logic gates.

Aim: To Build a Feed Forward neural network for prediction of logic gates.

Definition: A Feed forward Neural Network (FFNN) is a type of neural network where data flows in one direction—from input to output. It can be used to predict logic gate outputs (like AND, OR, XOR) by learning from input-output examples using layers of neurons and activation functions.

Source code:

```
import numpy as np
# XOR input and output
X = np.array([[0, 0],
              [0, 1],
              [1, 0],
              [1, 1]])
y = np.array([[0], [1], [1], [0]])  # Output
from tensorflow import keras
from tensorflow.keras import layers
# Create the model
model = keras.Sequential([
    layers.Dense(2, activation='relu', input_shape=(2,)),  # Hidden layer with 2 neurons
    layers.Dense(1, activation='sigmoid')  # Output layer with 1 neuron
])
# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
# Train the model
model.fit(X, y, epochs=5000, verbose=0)  # Increase epochs if necessary
# Evaluate the model
loss, accuracy = model.evaluate(X, y)
print(f'Loss: {loss}, Accuracy: {accuracy}')
```

```
# Make predictions
Z = np.array([[0, 1],
              [0, 1],
              [1, 0],
              [1, 1]])

predictions = model.predict(Z)
print("Predictions:")
print(np.round(predictions))  # Round to get binary output
```

Result:

```
Loss: 0.48011985421180725, Accuracy: 0.75

Predictions:
[[1.]
 [1.]
 [1.]
 [0.]]
```

Data Analytics using R- Lab

III. B. Tech I Semester

Prepared by:
Dr.Sayyad Rasheeduddin
Associate Professor

2024-2025

LEAD EXPERIMENTS

1. **Write R program to find Correlation and Covariance**
2. **Write R program to build clustering model using K-mean algorithm**

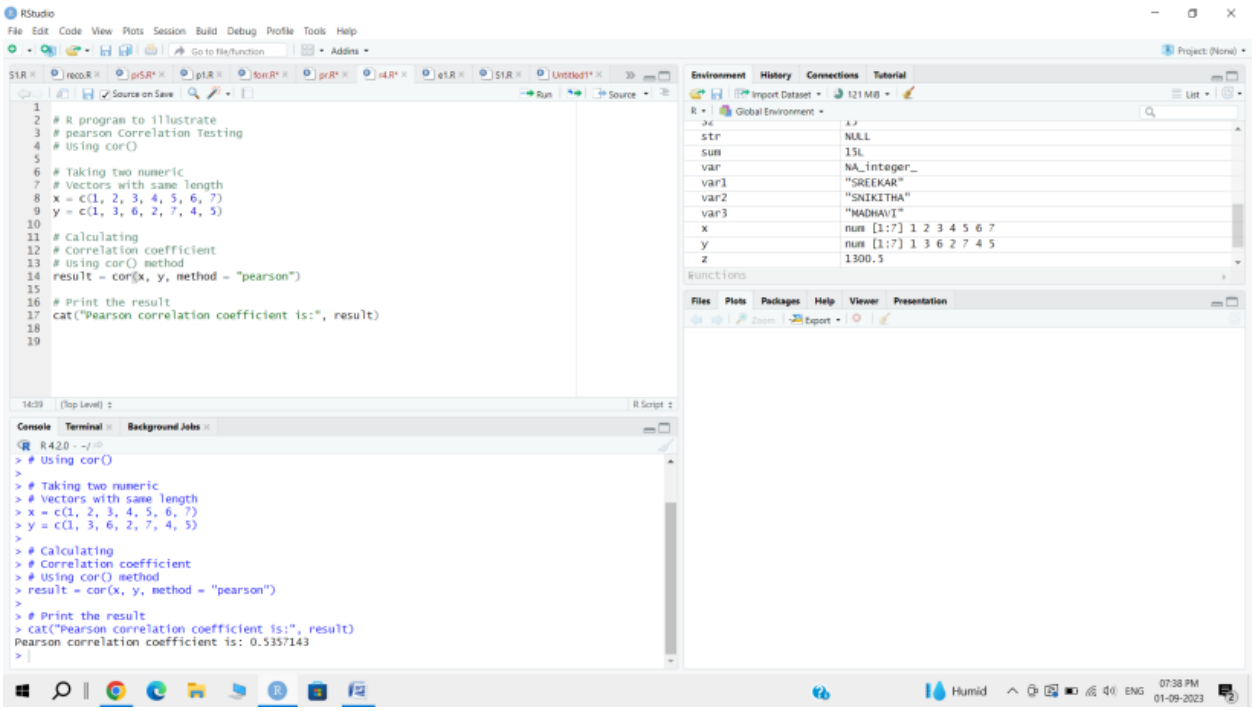
1. Write R program to find Correlation and Covariance

Program Description:
Covariance shows the direction of the path of the linear relationship between the variables while a function is applied to them.

Correlation on the contrary measures both the power and direction of the linear relationship between two variables.

```
# R program to illustrate
# pearson Correlation Testing
# Using cor()
# Taking two numeric
# Vectors with same length
x = c(1, 2, 3, 4, 5, 6, 7)
y = c(1, 3, 6, 2, 7, 4, 5)
# Calculating
# Correlation coefficient
# Using cor() method
result = cor(x, y, method = "pearson")
# Print the result
cat("Pearson correlation coefficient is:", result)
```

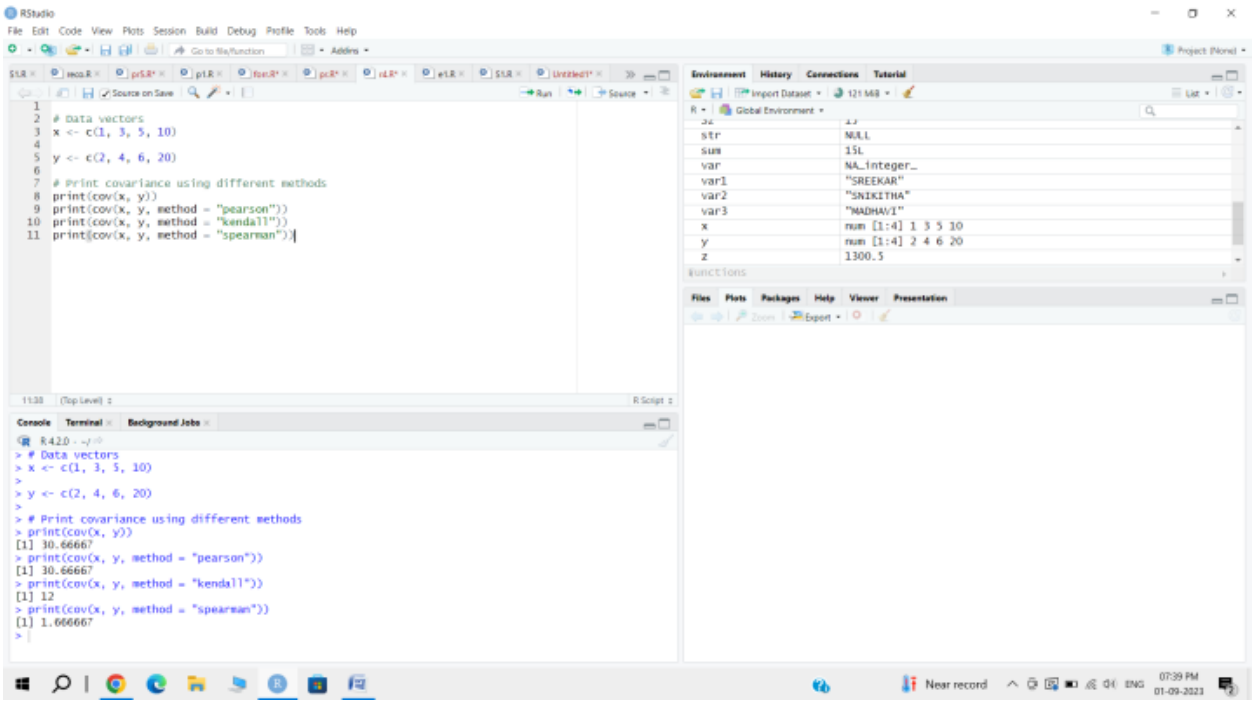
Output:



2. Covariance

```
# Data vectors
x <- c(1, 3, 5, 10)
y <- c(2, 4, 6, 20)
# Print covariance using different methods
print(cov(x, y))
print(cov(x, y, method = "pearson"))
print(cov(x, y, method = "kendall"))
print(cov(x, y, method = "spearman"))
```

Output:



2. **Write R program to build clustering model using K-mean algorithm**

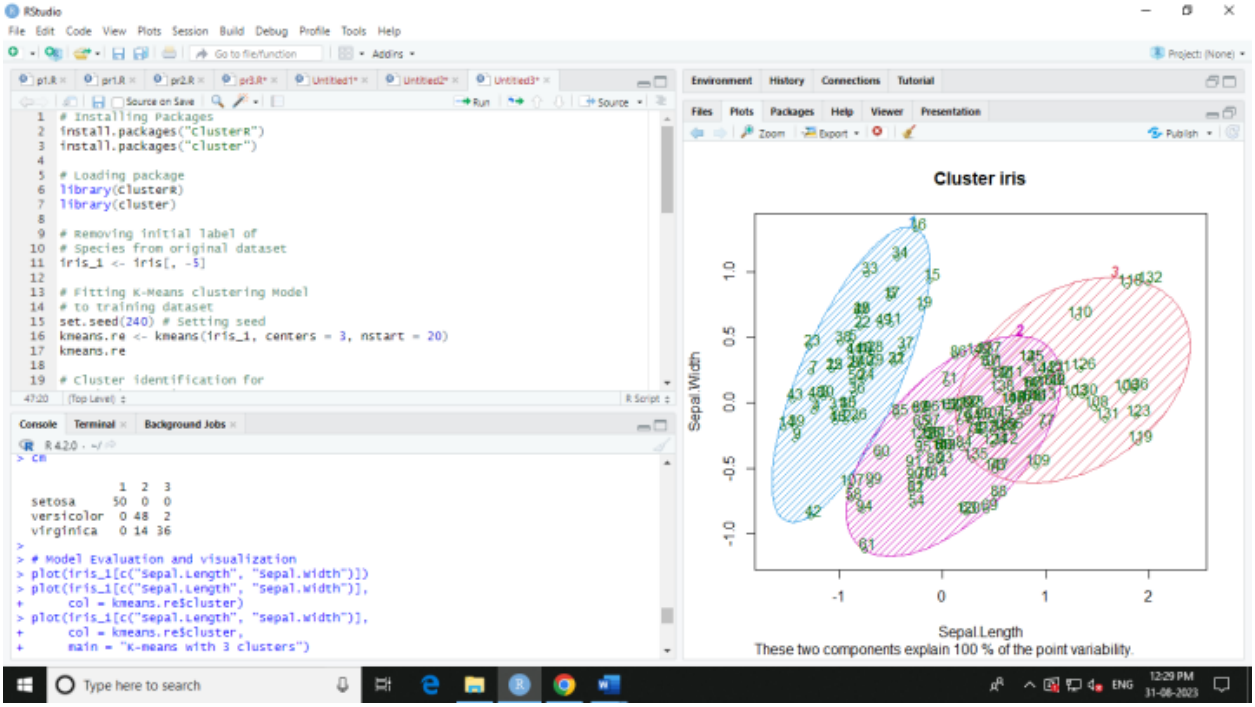
Program Description :

K Means Clustering in R Programming is an Unsupervised Non-linear algorithm that cluster data based on similarity or similar groups. It seeks to partition the observations into a pre-specified number of clusters. Segmentation of data takes place to assign each training example to a segment called a cluster

```
Loading data
data(iris)
# Structure
str(iris)
# Installing Packages
install.packages("ClusterR")
install.packages("cluster")
# Loading package
library(ClusterR)
library(cluster)
# Removing initial label of
# Species from original dataset
iris_1 <- iris[, -5]
# Fitting K-Means clustering Model
# to training dataset
set.seed(240) # Setting seed
kmeans.re <- kmeans(iris_1, centers = 3, nstart = 20)
kmeans.re
4647
# Cluster identification for
# each observation
kmeans.re$cluster
# Confusion Matrix
cm <- table(iris$Species, kmeans.re$cluster)
cm
# Model Evaluation and visualization
plot(iris_1[c("Sepal.Length", "Sepal.Width")])
plot(iris_1[c("Sepal.Length", "Sepal.Width")],
col = kmeans.re$cluster)
plot(iris_1[c("Sepal.Length", "Sepal.Width")],
col = kmeans.re$cluster,
main = "K-means with 3 clusters")
## Plotiing cluster centers
kmeans.re$centers
kmeans.re$centers[, c("Sepal.Length", "Sepal.Width")]
# cex is font size, pch is symbol
points(kmeans.re$centers[, c("Sepal.Length", "Sepal.Width")],
col = 1:3, pch = 8, cex = 3) 48
## Visualizing clusters
y_kmeans<- kmeans.re$cluster
clusplot(iris_1[, c("Sepal.Length", "Sepal.Width")],
y_kmeans,
lines = 0,
shade = TRUE,
color = TRUE,
labels = 2,
plotchar = FALSE,
span = TRUE,
```

main = paste("Cluster iris"),
xlab = 'Sepal.Length',
ylab = 'Sepal.Width')

Output:



Python Programming Lab

III B. Tech I Semester

SUHASINI T S
Asst. Professor

2024-2025

LEAD EXPERIMENTS

1) Write a python program to give input of a excel file and display the file on output console.
2) Write a python code that accepts the voice input and displays the voice message in text.

1) **Write a python program to give input of a excel file and display the file on output console.**

Aim: To give the input excel file and display the excel sheet.

Procedure:

Steps to Import an Excel File into Python using Pandas

Step 1: Capture the file path

First, you'll need to capture the full path where the Excel file is stored on your computer. For example, let's suppose that an Excel file is stored under the path

In the Python code, to be provided below, you'll need to modify the path name to reflect the location where the Excel file is stored on your computer.

Don't forget to include the file name (in our example, it's 'Product list' as highlighted in blue). You'll also need to include the Excel file extension

Step 2: Apply the Python code

And here is the Python code tailored to our example. Additional notes are included within the code to clarify some of the components used.

Step 3: Run the Python code to import the Excel file

You may then use the PIP install approach to install openpyxl for .xlsx files: pip install openpyxl

Code:

```
import pandas as pd

def display_excel_file(file_path):
    """
    Reads an Excel file and displays its content on the console.

    Args:
        file_path (str): The path to the Excel file.
    """
    try:
        # Read the Excel file into a pandas DataFrame
        df = pd.read_excel(file_path)
```

```
        # Display the DataFrame to the console
        print(f"Contents of '{file_path}':\n")
        print(df)

    except FileNotFoundError:
        print(f"Error: The file '{file_path}' was not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Example usage:
if __name__ == "__main__":
       file_path =” D:/NBA 24-25/ column "

    try:
        pd.DataFrame({'Col A': [1, 2, 3], 'Col B': ['X', 'Y', 'Z']}).to_excel(file_path,
index=False)
        print(f"Created a dummy Excel file at '{file_path}' for demonstration.")
    except Exception as e:
        print(f"Could not create dummy file: {e}")

    display_excel_file(file_path)

)
```

Output:

Created a dummy Excel file at ' D:/NBA 24-25/column” for demonstration.

Contents of '/content/sample_data/your_excel_file.xlsx':

	Col A	Col B
0	1	X
1	2	Y
2	3	Z

2. Write a python code that accepts the voice input and displays the voice message in text format

Aim: To convert voice into text.

Code :

```
# Python program to translate
# speech to text and text to speech


import speech_recognition as sr
import pyttsx3

# Initialize the recognizer
r = sr.Recognizer()

# Function to convert text to
# speech
def SpeakText(command):

    # Initialize the engine
    engine = pyttsx3.init()
    engine.say(command)
    engine.runAndWait()


# Loop infinitely for user to
# speak

while(1):

    # Exception handling to handle
    # exceptions at the runtime
    try:
```

```
        # use the microphone as source for input.
        with sr.Microphone() as source2:

            # wait for a second to let the recognizer
            # adjust the energy threshold based on
            # the surrounding noise level
            r.adjust_for_ambient_noise(source2, duration=0.2)

            #listens for the user's input
            audio2 = r.listen(source2)

            # Using google to recognize audio
            MyText = r.recognize_google(audio2)
            MyText = MyText.lower()

            print(&quot;Did you say &quot;,MyText)
            SpeakText(MyText)

    except sr.RequestError as e:
        print(&quot;Could not request results; {0}&quot;.format(e))

    except sr.UnknownValueError:
        print(&quot;unknown error occurred&quot;)
```

Input: voice speech (Hi buddy how are you)

Output: Did you say hi buddy how are you