ఆంధ్రప్రదేశ్ కేంద్రీయ విశ్వవిద్యాలయం आंध्रप्रदेश केंद्रीय विश्वविद्यालय

CENTRAL UNIVERSITY OF ANDHRA PRADESH

(Established by an act of Parliament in 2019)

DEPARTMENT OF COMPUTER SCIENCE AND A.I

Introduction to R-Programming Lab for MSC(AI&DS), MSC(CS)&MSC(MAT)

LAB MANUAL

Prepared by
D.Mahaboob Basha
Faculty
Department of Computer Science
Central University of Andhra Pradesh
Ananthapuramu.

INDEX

1.Write an R program that demonstrates the fundamental concepts of R programming, including data types, variables, operators, and other essential elements?

```
#Data Types :
numeric_var <- 10.5  # Numeric
integer_var <- 10L   # Integer
character_var <- "Hello, R!"  # Character
logical_var <- TRUE  # Logical
complex_var <- 3 + 4i  # Complex
date_var <- as.Date("2025-03-01")  # Date

# Print Data Types
print(class(numeric_var))
print(class(integer_var))
print(class(character_var))
print(class(logical_var))
print(class(complex_var))
print(class(date_var))
```

Vectors :When you want to create vector with more than one element, you should use c() function which means to combine the elements into a vector.

```
 # Create a vector.
apple <- c('red','green',"yellow")
print(apple)
# Get the class of the vector.
print(class(apple))
```

Lists

A list is an R-object which can contain many different types of elements inside it like vectors, functions and even another list inside it.

```
# Create a list.
list1 <- list(c(2,5,3),21.3,sin)
# Print the list.
print(list1)
```

Matrices

A matrix is a two-dimensional rectangular data set. It can be created using a vector input to the matrix function.

```
# Create a matrix.
M =matrix( c('a','a','b','c','b','a'),nrow=2,ncol=3,byrow= TRUE)
 print(M)
```

Arrays

While matrices are confined to two dimensions, arrays can be of any number of dimensions. The array function takes a dim attribute which creates the required number of dimension. In the below example we create an array with two elements which are 3x3 matrices each.

```
# Create an array.
a <- array(c('green','yellow'),dim= c(3,3,2))
 print(a)
```

Variables: The variables can be assigned values using leftward, rightward and equal to operator. The values of the variables can be printed using print() or cat() function. The cat() function combines multiple items into a continuous print output.

```
# Assignment using equal operator.
var1=c(0,1,2,3)
# Assignment using leftward operator.
var2<- c("learn","R")
# Assignment using rightward operator.
c(TRUE,1)->var3
print(var1)
cat ("var1 is ",var1,"\n")
cat ("var2 is ",var2,"\n")
cat ("var3 is ",var3,"\n")
```

R Operators : Types of Operators

```
#Arithmetic Operators
v <- c( 2,5.5,6)
t <- c(8, 3, 4)
print(v+t)
#Relational Operators
v <- c(2,5.5,6,9)
t <- c(8,2.5,14,9)
 print(v>t)
#Logical Operators
v <- c(3,1,TRUE,2+3i)
t <- c(4,1,FALSE,2+3i)
print(v&t)
#Assignment Operators
v1 <- c(3,1,TRUE,2+3i)
v2 <<- c(3,1,TRUE,2+3i)
v3 = c(3,1,TRUE,2+3i)
print(v1)
print(v2)
print(v3)
```

2. Write a R program code to perform different operations on matrices?

```
A=matrix(c(1:9),nrow=3,ncol=3,byrow=T)

B=matrix(c(10:18),nrow=3,ncol=3,byrow=T)

m1=nrow(A)

n1=ncol(A)
```

```
m2=nrow(B)
n2=ncol(B)
cat("Matrix A:\n")
print(A)
cat("Matrix B:\n")
print(B)
if(m1==m2 && n1==n2)
{
 cat("Sum of the matrices is A+B=\n")
 print(A+B)
} else
 cat("\n Addition of matrices is not possible")

if(n1==m2)
{
 cat("Product of the matrices is A*B=\n")
 print(A%*%B)
} else
cat("\n Multiplication of matrices is not possible")
cat("Transpose of the Matrix A is:\n")
print(t(A))
cat("Transpose of the Matrix B is:\n")
print(t(B))
```

3.Write a R Program code to create a list containing a vector, a matrix and a list and write a code for the following.
 # 1) Give names to the elements in the list
 # 2) Add element at the end of the list
 # 3) Remove the second element
Code:

```
#Creating a list
a=c(23,4,5,56)
b=matrix(data=1:9,nrow=3)
c=list(35,"Sai","Male")
lst=list(a,b,c)
```

```
print(lst)
# Giving names to the elements
print("Give names to the elements:")
names(lst)=c("vector", "matrix", "info")
print(lst)
# Adding element at the end of the list
print("Add element at the end of the list:")
lst[[4]]=c(1,2,3)
print(lst)
# Removing the second element of the list
cat("After removing the second element the list is:\n")
lst[[2]]=NULL
print(lst)
```

4.Write a R program to create a data frame from four given vectors.

```
name = c('Nandini','Joy','Snigdha','Chandana')
score= c(12.5,9,16.5,12,)
attempts= c(1,3,2,3)
qualify = c('yes','no','yes','no')
print("Original data frame:")
df=data.frame(name, score, attempts, qualify)
print(df)
```

5. R program to create a data frame of student with four given vectors(rno,name,grade,gendor) and write a code

```
# 1) to get the structure of a given data frame.
# 2) to get the statistical summary and nature of the data of a given data frame.
# 3) to extract specific column from a data frame using column name.
# 4) to extract first two rows from a given data frame.
 # 5) to extract 3rd and 5th rows with 1st and 3rd columns from a given data frame.
 # 6) to add a new column in a given data frame.
# 7) to add new row(s) to an existing data frame.
 # 8) to drop column(s) by name from a given data frame.
# 9) to drop row(s) by number from a given data frame.
# 10) to extract the records whose grade is greater than 9.
```

```
#Data frame creation
rno = c("24MAI001", "24MAI002", "24MAI003", "24MAI004")
name = c("Vijay", "Raghu", "Latha", "Mamatha")
grade = c(8.4, 9.9, 7.5, 9.1)
gendor = c("Male", "Male", "Female", "Female")
studf = data.frame(rno, name, grade, gendor)
print(studf)  # Corrected variable name
# 1) Getting the structure of data frame
print("The structure of the data frame is :")
print(str(studf))
# 2) Statistical summary and nature of the data
print("The statistical summary and nature of the data is :")
print(summary(studf))
#3)Extracting the Column heading name
print("The List of names in the column 'name' are")
print(studf$name)
4)#Extracting first two rows of dataframe
print("The first two rows of the dataframe are")
print(studf[1:2,])
```

6.Implement R-Loops with different examples?

```
# Create states vector
states <- c('Andhra pradesh','kerala','Rajastan','Telangana')
# Create the for statement
for ( i in states)
{
print(i)
}
# Creating a matrix
mat <- matrix(data = seq(10, 21, by=1), nrow = 6, ncol =2)
# Creating the loop with r and c to iterate over the matrix
for (r in 1:nrow(mat))
for (c in 1:ncol(mat))
print(paste("mat[", r, ",",c, "]=", mat[r,c]))
```

```
print(mat)
```

#using While Loop

```
v <- c("CUAP",”Hello CUAP”)
cnt <- 2
while (cnt < 7)
{
print(v)
cnt = cnt + 1
}
```

7.Write a R program to find the factorial of given number using recursive function calls.

```
# Recursive function to calculate factorial
factorial <- function(n) {
  if (n == 0 || n == 1) {
    return(1)
  } else {
    return(n * factorial(n - 1))
  }
}
# Get user input for the number
num <- as.integer(readline("Enter a number to calculate factorial: "))
# Calculate factorial using the recursive function
result <- factorial(num)
# Display the result
cat("Factorial of", num, "is:", result, "\n")
```

Sample output:

```
Enter a number to calculate factorial: 5
Factorial of 5 is: 120
```

8. Write a R program to find Sum, Mean and Product of a Vector, ignore

element like NA or NaN.

```
x = c(10, NULL, 20, 30, NA)
print("Sum:")
print(sum(x, na.rm=TRUE))
```

```
print("Mean:")
print(mean(x, na.rm=TRUE))
print("Product:")
print(prod(x, na.rm=TRUE))
```

9. Write a R program to save the information of a data frame in a file and display the information of the file.

```
cricket_data = data.frame(
name = c('Rohit','Rahul','Virat','Shreyas','Surya','Risabh','Iyer',
'bishnoi','shami','Buvi','Bumrah'),
  avg = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 0, 13.5, 8, 19),
  age = c(31, 29, 33, 27, 25, 26, 24, 21, 29, 30, 29),
  bowl = c('yes','no','yes','no','no','yes','yes','yes','yes','yes','yes'))
print("Original dataframe:")
print(cricket_data)
write.csv(cricket_data, file = "my_cricket_team1.csv")
print(paste("Please go to the location and open and check file contents", getwd()))
```

10.Write a R program for any visual representation of an object with creating graphs using graphic functions: Plot( ), Hist( ), Linechart( ), Pie( ), Boxplot( ), Scatterplots( ).

```
# Create a sample dataset
data <- c(23, 45, 56, 32, 67, 89, 55, 43, 78, 36, 49, 60, 70)
# Create a basic line chart
plot(data, type = "l", col = "blue", xlab = "X-axis", ylab = "Y-axis", main = "Line Chart")

# Create a histogram
hist(data, col = "lightblue", xlab = "Value", ylab = "Frequency", main = "Histogram")

# Create a pie chart
pie_data <- c(20, 30, 40, 10)
pie(pie_data, labels = c("A", "B", "C", "D"), col = rainbow(length(pie_data)), main = "Pie
Chart")
```

```
# Create a boxplot
boxplot(data, col = "lightgreen", xlab = "Value", main = "Box Plot")
# Create a scatterplot
x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
y <- c(2, 4, 5, 7, 8, 10, 11, 14, 15, 17)
plot(x, y, col = "red", xlab = "X-axis", ylab = "Y-axis", main = "Scatterplot")
# Add a regression line to the scatterplot
abline(lm(y ~ x), col = "blue")
# Create a legend for the scatterplot
legend("topleft", legend = "Regression Line", col = "blue", lty = 1)
```