1. *R Program for Different Data Structures*

R

```
# Vector
character_vector <- c("apple", "banana", "cherry")
character_vector

# Matrix
numeric_matrix <- matrix(1:6, nrow = 2, ncol = 3)
numeric_matrix

# List
my_list <- list(
  name = c("John", "Daniel", "Jack"),
  age = c(30, 53, 40),
  hobbies = c("reading", "golf", "gaming")
)
my_list

# Data Frame
data_frame <- data.frame(
  Name = c("Alice", "Bennett", "Charlie"),
  Age = c(25, 30, 22),
  Gender = c("Female", "Male", "Male")
)
data_frame

# Factor
gender <- c("Male", "Female", "Male", "Female", "Male")
factor_gender <- factor(gender, levels = c("Male", "Female"))
factor_gender
```

```
# Array
arr <- array(1:24, dim = c(4, 3, 2))
arr
```

---

2. *Variables, Constants, Data Types*

R

```
# Variables
radius <- 5
radius
name <- "Alice"
name
age <- 30L
age
is_student <- TRUE
is_student

# Constants
PI <- 3.14159265359
paste("Constant Value:", PI)
GREETING <- "Hello, World!"
paste("Constant Value:", GREETING)

# Data Types
print(class(radius))
print(class(name))
print(class(age))
print(class(is_student))
```

---

3. *Operators, Control Structures, Default Values, Complex Objects*

R

```
# Arithmetic Operators
a <- 11
b <- 4
sum_result <- a + b
sum_result
diff_result <- a – b
diff_result
product_result <- a * b
product_result
division_result <- a / b
division_result
modulus_result <- a %% b
modulus_result

# If-Else
if (a > b) {
  print("a is greater than b")
} else if (a < b) {
  print("a is less than b")
} else {
  print("a is equal to b")
}

# Default Arguments
my_function <- function(country = "INDIA") {
  paste("I am from", country)
```

```
}
my_function("USA")
my_function()

# Returning Complex Objects
res <- function() {
  v <- c(1, 2, 5, 3, 8)
  m <- matrix(1:8, ncol = 4)
  v_mean <- mean(v)
  m_min <- min(m)
  list(vec = v_mean, mat = m_min)
}
res()
```

---

5. *Cumulative Sums, Minima, Maxima, Calculus*

R

```
numbers <- c(1, 2, 3, 4, 5)
cumulative_sum <- cumsum(numbers)
cumulative_sum
cumulative_product <- cumprod(numbers)
cumulative_product
min_value <- min(numbers)
min_value
max_value <- max(numbers)
max_value

library(Deriv)
f <- function(x) x^2
```

```
derivative <- Deriv(f)
integral <- integrate(f, lower = 1, upper = 5)
integral
```

---

6. *Stationary Distribution of Markov Chains*

R

```
library(markovchain)
transition_matrix <- matrix(c(0.8, 0.2, 0.4, 0.6), nrow = 2, byrow = TRUE)
states <- c("State A", "State B")
my_markov_chain <- new("markovchain", states = states, transitionMatrix = transition_matrix)
steadyStates(my_markov_chain)
```

---

7. *Linear Algebra Operations*

R

```
matrix_A <- matrix(1:4, nrow = 2)
matrix_B <- matrix(3:8, nrow = 2)
det(matrix_A)
solve(matrix_A)
 vector_a <- c(2, 4, 6)
vector_b <- c(1, 2, 5)
vector_sum <- vector_a + vector_b
matrix_product <- matrix_A %*% matrix_B
A <- matrix(c(2, 1, 1, 7), nrow = 2)
 b <- c(5, 7)
# Solve for x
x <- solve(A, b)
cat("Solution of the linear equation Ax = b:\n")
print(x)
```

---

8. *Visual Representations*

R

```
data <- c(3, 4, 7, 8, 9, 10, 12, 14, 15, 18, 21)

plot(data)

hist(data, breaks = 5, main = "Histogram",xlab=”value”,ylab=”frequency”, col = "green")

line_data <- cumsum(data)

plot(1:length(data), line_data, type = "l", col = "red", main = "Line Chart", xlab=”value”,ylab=”frequency”)

slices <- c(30, 20, 10, 40)

lbls <- c("Slice 1", "Slice 2", "Slice 3", "Slice 4")

pie(slices, labels = lbls, main = "Pie Chart")

boxplot(data, main = "Boxplot", col = "purple",main=”Boxplot”)
```

---

9. *Data Manipulation*

R

```
data_frame <- data.frame(
  Name = c("Alice", "Bennett", "Charlie", "David", "Emma"),
  Age = c(25, 30, 22, 28, 35),
  Gender = c("Female", "Male", "Male", "Male", "Female"),
  Score = c(85, 92, 78, 88, 95)
)
subset_data <- data_frame[data_frame$Age > 25, ]
subset_data
summary_stats<-summary(data_frame$Score)
summary_stats
data_frame$Grade <- ifelse(data_frame$Score >= 90, "A", ifelse(data_frame$Score >= 80, "B", "C"))
gen<-aggregate(data_frame$Score ,by=list(data_frame$Gender),FUN=mean)
colnames(gender_avg_score)<-c(“Gender”,”Avg_Score”)
print(gender_avg_score)
```
