1. **Write a R program to take input from the user (name and age) and display the values. Also print the version of R installation.**

```
# Taking input from user
name <- readline(prompt = "Enter your name: ")
age <- as.numeric(readline(prompt = "Enter your age: "))

# Displaying input values
cat("Your name is:", name, "\n")
cat("Your age is:", age, "\n")

# Printing the version of R installation
cat("R Version Installed:\n")
print(R.version.string)
```

2. **Using R import the data from Excel/.CSV file and find mean, median, mode, quartiles.**

Using .CSV file:

```
library(modeest)
data <- read.csv("data.csv")
numeric_column <- data[[1]]

mean_val <- mean(numeric_column, na.rm = TRUE)
median_val <- median(numeric_column, na.rm = TRUE)
mode_val <- mfv(numeric_column, na.rm = TRUE)
quartiles <- quantile(numeric_column, na.rm = TRUE)

cat("Mean:", mean_val, "\n")
cat("Median:", median_val, "\n")
cat("Mode:", mode_val, "\n")
cat("Quartiles:\n")
print(quartiles)
```

1. **Write a R program to create a simple bar plot of five subjects marks.**

```
subjects <- c("Math", "Science", "English", "History", "Computer")
marks <- c(85, 78, 92, 74, 88)
barplot(marks,
        names.arg = subjects,
        col = "skyblue",
        main = "Marks in Five Subjects",
        xlab = "Subjects",
        ylab = "Marks",
        ylim = c(0, 100))
```

2. **Using R import the data from Excel/.CSV file and find standard deviation, variance and co-variance.**

```
data <- read.csv("data.csv")  # Replace with your actual filename
x <- data[[2]]  # You can also use data$ColumnName
y <- data[[3]]

std_dev_x <- sd(x, na.rm = TRUE)
variance_x <- var(x, na.rm = TRUE)
covariance_xy <- cov(x, y, use = "complete.obs")

cat("Standard Deviation of X:", std_dev_x, "\n")
cat("Variance of X:", variance_x, "\n")
cat("Covariance between X and Y:", covariance_xy, "\n")
```

1. **Write a R program to create a list containing a vector, a matrix and a list and remove the second element.**

```
my_vector <- c(1, 2, 3)
my_matrix <- matrix(1:9, nrow = 3)
my_inner_list <- list("a", "b", "c")
my_list <- list(my_vector, my_matrix, my_inner_list)
cat("Original List:\n")
print(my_list)
my_list[[2]] <- NULL
cat("\nList After Removing Second Element:\n")
print(my_list)
```

3. **Write a R program to call the (built-in) dataset air quality. Remove the variables 'Solar.R' and 'Wind' and display the data frame.**

```
data("airquality")
cat("Original Column Names:\n")
print(colnames(airquality))
modified_data <- subset(airquality, select = -c(Solar.R, Wind))

cat("\nData Frame After Removing 'Solar.R' and 'Wind':\n")
print(head(modified_data))  # Print first few rows for brevity
```

Write a R program to create an empty data frame.

```
empty_df <- data.frame()
cat("Empty Data Frame:\n")
print(empty_df)


empty_df <- data.frame(ID=integer(),
                       Name=character(),
                       Salary=double(),
                       stringsAsFactors=FALSE)
print(empty_df)
```

3. **Write an R program to create a Data frames which contain details of 5employees and display the details in ascending order.**

```
employees <- data.frame(
 ID = c(105, 102, 101, 104, 103),
 Name = c("Alice", "Bob", "Charlie", "David", "Eva"),
 Salary = c(50000, 60000, 55000, 52000, 58000)
)
cat("Original Employee Details:\n")
print(employees)


sorted_employees <- employees[order(employees$ID), ]


cat("\nEmployee Details in Ascending Order (by ID):\n")
print(sorted_employees)
```

1. Write a R program to merge two given lists into one list.

```
list1 <- list("Apple", 42, TRUE)

list2 <- list("Banana", 99.5, FALSE)

merged_list <- c(list1, list2)


cat("Merged List:\n")

print(merged_list)
```

2. Consider Weather dataset i) Selecting using the column number ii) Selecting using the column nameiii) Make a scatter plot to compare Wind speed and temperature.

i) **Selecting Columns Using Column Number**

```
data("airquality")


selected_by_number <- airquality[, c(1, 3)]


cat("Columns selected by number (1st and 3rd):\n")

print(head(selected_by_number))
```

Selecting Columns Using Column Name

```
selected_by_name <- airquality[, c("Ozone", "Temp")]

cat("Columns selected by name (Ozone and Temp):\n")

print(head(selected_by_name))
```

Scatter Plot Comparing Wind Speed and Temperature

```
plot(airquality$Wind, airquality$Temp,

   main = "Scatter Plot: Wind Speed vs Temperature",

   xlab = "Wind Speed (mph)",

   ylab = "Temperature (F)",

   col = "blue",

   pch = 19)
```

1. Write a R program to get the unique elements of a given string and unique numbers of vector.

```
input_string <- "statistics"

unique_chars <- unique(strsplit(input_string, "")[[1]])

cat("Unique characters in the string:\n")

print(unique_chars)


num_vector <- c(1, 2, 2, 3, 4, 4, 5, 1)

unique_numbers <- unique(num_vector)

cat("\nUnique numbers in the vector:\n")

print(unique_numbers)
```

2. Write a R program to compare two data frames to find the row(s) in first data frame that are not present in second data frame.

```
df1 <- data.frame(ID = c(1, 2, 3, 4),

           Name = c("Alice", "Bob", "Charlie", "David"))


df2 <- data.frame(ID = c(2, 4),

           Name = c("Bob", "David"))

difference <- setdiff(df1, df2)


cat("Rows in df1 not present in df2:\n")

print(difference)
```

1. **Write R program to find whether given number is positive or negative.**

```
num <- as.numeric(readline(prompt = "Enter a number: "))

if (num > 0) {
  cat("The number is positive.\n")
} else if (num < 0) {
  cat("The number is negative.\n")
} else {
  cat("The number is zero.\n")
}
```

2. **Write R program to read number and print corresponding day name in a week.**

```
day_num <- as.integer(readline(prompt = "Enter a number (1 to 7): "))

days <- c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")


if (day_num >= 1 && day_num <= 7) {

  cat("The day is:", days[day_num], "\n")

} else {

  cat("Invalid input! Please enter a number between 1 and 7.\n")

}
```

Create a Matrix using R and Perform the operations addition, subtraction, multiplication.

```
matrix1 <- matrix(c(1, 2, 3, 4), nrow = 2)

matrix2 <- matrix(c(5, 6, 7, 8), nrow = 2)

cat("Matrix 1:\n")

print(matrix1)

cat("Matrix 2:\n")

print(matrix2)

add_result <- matrix1 + matrix2

cat("\nAddition of Matrices:\n")

print(add_result)

sub_result <- matrix1 - matrix2

cat("\nSubtraction of Matrices:\n")

print(sub_result)

mul_result <- matrix1 * matrix2

cat("\nElement-wise Multiplication of Matrices:\n")

print(mul_result)
```

3. **Write a R program to create an ordered factor from data consisting of the names of months.**

```
months_data <- c("March", "January", "April", "February", "December", "October")


ordered_months <- factor(months_data,

                 levels = c("January", "February", "March", "April", "May", "June",

                        "July", "August", "September", "October", "November", "December"),

                 ordered = TRUE)


cat("Ordered Factor of Months:\n")

print(ordered_months)
```

1. **Write a R program to find nth highest value in a given vector.**

```
vec <- c(55, 23, 89, 12, 75, 60)

n <- as.integer(readline(prompt = "Enter the value of n: "))


if (n <= length(vec)) {

  nth_highest <- sort(vec, decreasing = TRUE)[n]

  cat(n, "th highest value is:", nth_highest, "\n")

} else {

  cat("n is larger than the length of the vector.\n")

}
```

2. **Write a script in R to create a list of students and perform the following Give names to the students in the list. Add a student at the end of the list. Remove the first Student. Update the second last student.**

```
students <- list("Alice", "Bob", "Charlie", "David")

names(students) <- c("student1", "student2", "student3", "student4")

cat("Original list:\n")

print(students)

students[["student5"]] <- "Eva"

cat("\nAfter adding a student at the end:\n")

print(students)


students <- students[-1]

cat("\nAfter removing the first student:\n")

print(students)

second_last_index <- length(students) - 1

students[[second_last_index]] <- "Updated_Student"

cat("\nAfter updating the second last student:\n")

print(students)
```

1. Write an R program to sort a Vector in ascending and descending order.

```
vec <- c(23, 5, 12, 89, 34, 7)


ascending <- sort(vec)

cat("Ascending order:\n")

print(ascending)

descending <- sort(vec, decreasing = TRUE)

cat("Descending order:\n")

print(descending)
```

3. **Write an R Program to calculate Decimal into binary of a given number. Consider the inbuilt iris dataset Create a variable “y” and attach to it the output attribute of the “iris”dataset . Create a barplot to breakdown your output attribute. Create a density plot matrix for each attribute by class value**

```
decimal <- as.integer(readline(prompt = "Enter a decimal number: "))

binary <- rev(as.integer(intToBits(decimal)))
binary <- binary[which.max(binary):length(binary)]

cat("Binary equivalent:\n")
print(binary)
```

1. Write an R program to extract first 10 English letter in lower case and last 10 letters in upper case and extract letters between 22nd to 24th letters in upper case.

```
first_10_lower <- letters[1:10]
cat("First 10 lowercase letters:\n")
print(first_10_lower)

last_10_upper <- LETTERS[17:26]
cat("\nLast 10 uppercase letters:\n")
print(last_10_upper)
letters_22_24_upper <- LETTERS[22:24]
cat("\n22nd to 24th uppercase letters:\n")
print(letters_22_24_upper)
```

2. Write a script in R to create a list of students and perform the following Give names to the students in the list. Add a student at the end of the list. Remove the first Student. Update the second last student.

```
students <- list("Aarav", "Bhavya", "Chirag", "Divya")
names(students) <- c("student1", "student2", "student3", "student4")
cat("Original Student List:\n")
print(students)

students[["student5"]] <- "Esha"
cat("\nAfter adding a student at the end:\n")
print(students)
students <- students[-1]
cat("\nAfter removing the first student:\n")
print(students)
second_last_index <- length(students) - 1
students[[second_last_index]] <- "Updated_Student"
cat("\nAfter updating the second last student:\n")
print(students)
```

1. Write an R program to convert a given matrix to a list and print list in ascending order.

```
m <- matrix(c(12, 5, 8, 19, 3, 7), nrow = 2)

lst <- as.list(m)

sorted_lst <- sort(unlist(lst))

print(sorted_lst)
```

2. Consider Weather dataset i) Selecting using the column number ii) Selecting using the column nameiii) Make a scatter plot to compare Wind speed and temperature.

```
weather <- data.frame(
  Temp = c(25, 28, 31, 22, 30),
  Wind = c(10, 8, 12, 9, 7),
  Humidity = c(80, 75, 70, 90, 85)
)


weather[, 2]

weather$Wind


plot(weather$Wind, weather$Temp, main = "Wind Speed vs Temperature",
     xlab = "Wind Speed", ylab = "Temperature", col = "blue", pch = 19)
```

1. Write a R program to create a data frame using two given vectors and display the duplicated elements and unique rows of the said data frame.

```
names <- c("Amit", "Neha", "Raj", "Neha", "Amit")
marks <- c(85, 90, 78, 90, 85)

df <- data.frame(Name = names, Marks = marks)
print("Original Data Frame:")
print(df)
dup_rows <- df[duplicated(df), ]
print("Duplicated Rows:")
print(dup_rows)
unique_rows <- unique(df)
print("Unique Rows:")
print(unique_rows)
```

2. Write an R Program to calculate Decimal into binary of a given number. Consider the inbuilt iris dataset i) Create a variable "y" and attach to it the output attribute of the "iris"dataset .ii) Create a barplot to breakdown your output attribute. iii) Create a density plot matrix for each attribute by class value.

```
decimal <- as.integer(readline(prompt = "Enter a decimal number: "))
binary <- rev(as.integer(intToBits(decimal)))
binary <- binary[which.max(binary):length(binary)]
cat("Binary equivalent is:\n")
print(binary)
data(iris)

# i) Create a variable y with the output attribute (Species)
y <- iris$Species

# ii) Create a barplot of the output attribute
barplot(table(y),
    main = "Species Count in Iris Dataset",
    col = c("lightblue", "lightgreen", "lightpink"),
    xlab = "Species",
    ylab = "Count")
```

# iii) Density plot matrix by class

```
library(lattice)
densityplot(~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width | Species,
         data = iris,
         auto.key = TRUE,
         main = "Density Plot Matrix by Species")
```

1. Write a R program to multiply two vectors of integers type and length 3.

```
v1 <- c(2, 4, 6)
v2 <- c(3, 5, 7)
result <- v1 * v2
print(result)
```

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

```
df <- data.frame(Name = c("Amit", "Neha", "Raj"), Age = c(25, 23, 27))
write.csv(df, "datafile.csv", row.names = FALSE)
read_data <- read.csv("datafile.csv")
print(read_data)
```

1. Write a R program to list containing a vector, a matrix and a list and give names to the elements in the list.

```
vec <- c(10, 20, 30)

mat <- matrix(1:4, nrow = 2)

lst <- list("a" = 1, "b" = 2)

main_list <- list(Numbers = vec, Table = mat, Elements = lst)

print(main_list)
```

2. Write a R program to create a factor corresponding to height of women data set, which contains height and weights for a sample of women

```
data(women)

height_factor <- factor(women$height)

print(height_factor)
```