Practical No. 9 Date: / / 201

Aim: Program to perform data frame operations on Employee Details.

Objectives:

- To study R Data Frame & its operations.
- Implement a program to perform operations on Employee Details.

Theory:

R Data Frame

R is a statistical programming language, which often works with datasets. A dataset is consist of observations or instances associated with some variables. For example, dataset of people

name	age	Child
ABC	28	FALSE
DEF	30	TRUE
GHI	21	TRUE
JKL	39	FALSE
MNO	35	TRUE

We cannot store this information in either matrix or list, because of the limitations.

In R, the fundamental data structure to store typical datasets is Data Frame. A data frame similar to matrix consist of rows and columns, but can contain elements of different types. Each column contains values of one variable and each row contains values for observations or instances. Following are the characteristics of data frames:

- Elements in data frame can be numeric, character, factor, logical etc.
- Elements in a column should be of same type.
- The column names should be non-empty.
- The row names (if present) should be unique.
- Each column should contain same number of data items.

A data frame mostly created by importing data from:

- CSV file
- Relational database (e.g. SQL)
- Software Packages like Excel, SPSS etc.

To manually create data frame in R data.frame() function is used.

Creating Data Frame

Syntax

```
data.frame(dataframe_elements)
```

- dataframe_elements: vector/matrix/factor etc.

Example

```
name <- c("ABC","DEF","GHI","JKL", "MNO")
age <- c(28,30,21,39,35)
child <- c(FALSE,TRUE,TRUE,FALSE,TRUE)
people_df <- data.frame(name,age,child)
people_df
  name    age  child       # Column names coressponds to variable names
1 ABC     28   FALSE
2 DEF     30   TRUE
3 GHI     21   TRUE
4 JKL     39   FALSE
5 MNO     35   TRUE
```

Naming Data Frame Columns

Using names() Function

```
name <- c("ABC","DEF","GHI","JKL", "MNO")
age <- c(28,30,21,39,35)
child <- c(FALSE,TRUE,TRUE,FALSE,TRUE)
people_df <- data.frame(name,age,child)
names(people_df) <- c("Name","Age","Child")
people_df
  Name    Age  Child       # Updated Column names
1 ABC     28   FALSE
2 DEF     30   TRUE
3 GHI     21   TRUE
4 JKL     39   FALSE
5 MNO     35   TRUE
```

Using data.frame() Function

```
name <- c("ABC","DEF","GHI","JKL", "MNO")
age <- c(28,30,21,39,35)
child <- c(FALSE,TRUE,TRUE,FALSE,TRUE)
people_df <- data.frame(Name = name,Age = age,Child = child)
people_df
  Name    Age  Child        # With Column names
1 ABC     28   FALSE
2 DEF     30   TRUE
3 GHI     21   TRUE
4 JKL     39   FALSE
```

```
5 MNO     35    TRUE
```

Data Frame Structure

```
name <- c("ABC","DEF","GHI","JKL", "MNO")
age <- c(28,30,21,39,35)
child <- c(FALSE,TRUE,TRUE,FALSE,TRUE)
people_df <- data.frame(Name = name,Age = age,Child = child,
                        stringsAsFactor = FALSE)
str(people_df)                      # String stored as string
'data.frame':   5 obs. of  3 variables:
 $ Name : chr  "ABC" "DEF" "GHI" "JKL" ...
 $ Age  : num  28 30 21 39 35
 $ Child: logi  FALSE TRUE TRUE FALSE TRUE
```

Accessing Data Frame

The structure of data frame is intersection between matrix & list. So, accessing data frame element(s) needs syntax from both,

- To access elements as matrix "[" is used.
- To access elements as list "[[" or "$" is used.

Like any R object, data frames are also accessed by index or by name of element.
As matrix using [

```
name <- c("ABC","DEF","GHI","JKL", "MNO")
age <- c(28,30,21,39,35)
child <- c(FALSE,TRUE,TRUE,FALSE,TRUE)
people_df <- data.frame(Name = name,Age = age,Child = child,
                        stringsAsFactor = FALSE)

people_df[3,2]
people_df[3,"Age"]                  # Single element
[1] 21

people_df[3,]                       # Single Row
 Name     Age  Child
3 GHI     21   TRUE

people_df[,"Age"]                   # Column as vector
[1] 28 30 21 39 35

people_df[c(3,5), c("Age","Child")]# Multiple Rows
```

```
     Age  Child
3    21   TRUE
5    35   TRUE

people_df[2]
people_df["Age"]                    # Column only
     Age
1    28
2    30
3    21
4    39
5    35
```

As list using [[or $

```
name <- c("ABC","DEF","GHI","JKL", "MNO")
age <- c(28,30,21,39,35)
child <- c(FALSE,TRUE,TRUE,FALSE,TRUE)
people_df <- data.frame(Name = name,Age = age,Child = child,
                        stringsAsFactor = FALSE)

people_df$Age
people_df[["Age"]]
people_df[[2]]                      # Column as vector
[1] 28 30 21 39 35
```

Augmenting Data Frame

In R, existing data frame can be augmented by

- Adding column(s) to add new variable(s)
- Adding row(s) to add new observation(s)

To add new column(s) [[or $ or cbind() function can be used. To add new row(s) rbind() function is used. The column(s) to be added must contain same number of element(s) as existing columns(s). The row(s) to be added must have same structure as existing row(s).

Adding Columns: [[or $

```
name <- c("ABC","DEF","GHI","JKL", "MNO")
age <- c(28,30,21,39,35)
child <- c(FALSE,TRUE,TRUE,FALSE,TRUE)
people_df <- data.frame(Name = name,Age = age,Child = child,
                        stringsAsFactor = FALSE)
```

```
     Name Age Child
1    ABC  28  FALSE
2    DEF  30  TRUE
3    GHI  21  TRUE
4    JKL  39  FALSE
5    MNO  35  TRUE

height <- c(163, 177, 163, 162, 157)
people_df[[Height]] <- height
people_df$Height <- height

people_df                          # Adding "Height" Column
     Name Age Child Height
1    ABC  28  FALSE   163
2    DEF  30  TRUE    177
3    GHI  21  TRUE    163
4    JKL  39  FALSE   162
5    MNO  35  TRUE    157
```

Adding Columns: cbind() function

```
Weight <- c(74, 63, 68, 55, 56)
people_df <-cbind(people_df,Weight)

people_df                          # Adding “Weight" Column
     Name Age Child Height Weight
1    ABC  28  FALSE   163     74
2    DEF  30  TRUE    177     63
3    GHI  21  TRUE    163     68
4    JKL  39  FALSE   162     55
5    MNO  35  TRUE    157     56
```

Adding Rows: rbind() function

```
name <- c("ABC","DEF","GHI","JKL", "MNO")
age <- c(28,30,21,39,35)
child <- c(FALSE,TRUE,TRUE,FALSE,TRUE)
people_df <- data.frame(Name = name,Age = age,Child = child,
                        stringsAsFactor = FALSE)
```

```
     Name Age Child
1    ABC  28  FALSE
2    DEF  30  TRUE
3    GHI  21  TRUE
4    JKL  39  FALSE
5    MNO  35  TRUE

newRow <- c("PQR", 37, FALSE)
people_df <- rbind(people_df,newRow)

people_df                          # Adding New Row
     Name Age Child
1    ABC  28  FALSE
2    DEF  30  TRUE
3    GHI  21  TRUE
4    JKL  39  FALSE
5    MNO  35  TRUE
6    PQR  37  FALSE

newRow <- data.frame(Name = "PQR", Age = 37, Child = FALSE)
people_df <- rbind(people_df,newRow)

people_df                          # Adding New Row
     Name Age Child
1    ABC  28  FALSE
2    DEF  30  TRUE
3    GHI  21  TRUE
4    JKL  39  FALSE
5    MNO  35  TRUE
6    PQR  37  FALSE
```

Data Frame Operations

Sorting: order() function

```
name <- c("ABC","DEF","GHI","JKL", "MNO")
age <- c(28,30,21,39,35)
child <- c(FALSE,TRUE,TRUE,FALSE,TRUE)
people_df <- data.frame(Name = name,Age = age,Child = child,
                        stringsAsFactor = FALSE)
```

```
     Name Age Child
1    ABC  28  FALSE
2    DEF  30  TRUE
3    GHI  21  TRUE
4    JKL  39  FALSE
5    MNO  35  TRUE

ranks <- order(people_df$Age)# Sort Elements & Returns ordered indices
ranks
[1] 3 1 2 5 4                # Corresponding to "Age": 21 28 30 35 39
people_df[ranks,]            # Using ordered indices to order data frame
    Name Age Child
3   GHI  21  TRUE
1   ABC  28  FALSE
2   DEF  30  TRUE
5   MNO  35  TRUE
4   JKL  39  FALSE

people_df[order(people_df$Age, decreasing = TRUE),]
                             # Sorting in decending order directly
    Name Age Child
4   JKL  39  FALSE
5   MNO  35  TRUE
2   DEF  30  TRUE
1   ABC  28  FALSE
3   GHI  21  TRUE
```

summary() function

```
name <- c("ABC","DEF","GHI","JKL", "MNO")
age <- c(28,30,21,39,35)
child <- c(FALSE,TRUE,TRUE,FALSE,TRUE)
people_df <- data.frame(Name = name,Age = age,Child = child,
                        stringsAsFactor = FALSE)
     Name Age Child
1    ABC  28  FALSE
2    DEF  30  TRUE
3    GHI  21  TRUE
4    JKL  39  FALSE
5    MNO  35  TRUE
```

```
summary(people_df)
 Name               Age             Child
 Length:5           Min.   :21.0    Mode :logical
 Class :character   1st Qu.:28.0    FALSE:2
 Mode  :character   Median :30.0    TRUE :3
                    Mean   :30.6
                    3rd Qu.:35.0
                    Max.   :39.0
```

Algorithm

1. Start.
2. Create a data frame “employee”.
3. Name every column of data frame using any one method.
4. Read choice for data frame operations from menu as
    a. Access Column
    b. Access Elements
    c. Add Column
    d. Add Row
    e. Sort Elements
    f. Summary
    g. Display
5. As per choice perform data frame operations as
    a. If choice is “a”, access a column using [.
    b. If choice is “b”, access elements using [[or $.
    c. If choice is “c”, add new column using any method.
    d. If choice is “d”, add new row using any method.
    e. If choice is “e”, sort data frame elements using sort() function.
    f. If choice is “f”, show summary of data frame using summary() function.
    g. If choice is “f”, display contents of data frame.
6. Stop.