Assignment No	7
Title	Advanced Datatypes in R
Objective	List, Matrix , Factor , Data Frame
Roll No	MCA2565

1) LIST

Source Code :-

```
mylist <- list(name = "hritik","age" = 20,city = "Mumbai")
print(mylist)

empid = c(1,2,3,4)
empName = c("xyz","ABC","DEF","PQR")
numberOfemp=4

empList <- list("ID" = empid, "Name" = empName, "ToTalEmp" = numberOfemp)
print(empList$Name)
print(empList$Name[2])
print(empList[2][1])

empid<-c(1,2,3,4)
empName <- c("hritik" , "ajay", "vinay")
numberOfemp <- 4
empList1<- list("ID" = empid, "Name" = empName, "TotalEmp"= numberOfemp)

empList1$'TotalEmp' = 5
empList1[[1]][5]=5
empList1[[2]][5]="KRISH"

print(empList1)

empList1[[2]][1]="RAM"
empList1$Name

#CONCATINATION
list1 = list(1,3,5,7,9)
list2 = list(2,4,6,8,10)

print(list1)
```

print(list2)

combined <- c(list1,list2)
print(combined)

Output :-

```
> mylist <- list(name = "hritik","age" = 20,city = "Mumbai")
> print(mylist)
$name
[1] "hritik"

$age
[1] 20

$city
[1] "Mumbai"

> empid = c(1,2,3,4)
> empName = c("xyz","ABC","DEF","PQR")
> numberOfemp=4
> empList <- list("ID" = empid, "Name" = empName, "ToTalEmp" = numberOfemp)
> print(empList$Name)
[1] "xyz" "ABC" "DEF" "PQR"
> print(empList$Name[2])
[1] "ABC"
> print(empList[2][1])
$Name
[1] "xyz" "ABC" "DEF" "PQR"

> empid<-c(1,2,3,4)
> empName <- c("hritik" , "ajay", "vinay")
> numberOfemp <- 4
> empList1<- list("ID" = empid, "Name" = empName, "TotalEmp"= numberOfemp)
> empList1$'TotalEmp' = 5
> empList1[[1]][5]=5
> empList1[[2]][5]="KRISH"
> print(empList1)
$ID
[1] 1 2 3 4 5

$Name
[1] "hritik" "ajay"   "vinay"  NA       "KRISH"

$TotalEmp
[1] 5

> empList1[[2]][1]="RAM"
> empList1$Name
[1] "RAM"   "ajay"  "vinay" NA      "KRISH"
> #CONCATINATION
> list1 = list(1,3,5,7,9)
> list2 = list(2,4,6,8,10)
```

```
> print(list1)
[[1]]
[1] 1

[[2]]
[1] 3

[[3]]
[1] 5

[[4]]
[1] 7

[[5]]
[1] 9

> print(list2)
[[1]]
[1] 2

[[2]]
[1] 4

[[3]]
[1] 6

[[4]]
[1] 8

[[5]]
[1] 10
```

```
> combined <- c(list1,list2)
> print(combined)
[[1]]
[1] 1

[[2]]
[1] 3

[[3]]
[1] 5

[[4]]
[1] 7

[[5]]
[1] 9

[[6]]
[1] 2

[[7]]
[1] 4

[[8]]
[1] 6

[[9]]
[1] 8

[[10]]
[1] 10
```

2) Matrix

Source Code :-

```
m <- matrix(c(11,12,13,55,60,65,66,72,78),nrow=3,ncol=3)
m

dim(m)

m <- matrix(c(11,12,13,55,60,65,66,72,78),nrow = 3,ncol = 3,byrow = TRUE)
m

#by default elements in the matrix are filled by col
#"byrow" attribute of matrix can be used
matrix(c(11,12,13,55,60,65,66,72,78),nrow =3 ,ncol=3,byrow=TRUE)
m

#cbind-ing and rbind-ing:
#By using cbind() and rbind() functions
x<-c(1,2,3)
y<-c(11,12,13)
```

```
z<-cbind(x,y)
z

z<-rbind(x,y)
z

p<-3*m
p

n<-matrix(c(4,5,6,14,15,16,24,25,26),nrow=3,ncol=3)
#Addition of two matrices
q<-m+n
q
o<-matrix(c(4,5,6,14,15,16,24,25,26),nrow=3,ncol=3)
o
#Matrix Multiplication by using %*%
r<-m %*% o
r
#Transpose of Matrix
mdash <- t(m)
mdash
s<-matrix(c(4,5,6,14,15,16,24,25,26),nrow=3,ncol=3,byrow=TRUE)
#Determinant of s
s_det<-det(s)
s_det
```

Output :-

```
> m <- matrix(c(11,12,13,55,60,65,66,72,78),nrow=3,ncol=3)
> m
     [,1] [,2] [,3]
[1,]   11   55   66
[2,]   12   60   72
[3,]   13   65   78
> dim(m)
[1] 3 3
> m <- matrix(c(11,12,13,55,60,65,66,72,78),nrow = 3,ncol = 3,byrow = TRUE)
> m
     [,1] [,2] [,3]
[1,]   11   12   13
[2,]   55   60   65
[3,]   66   72   78
> #by default elements in the matrix are filled by col
> #"byrow" attribute of matrix can be used
> matrix(c(11,12,13,55,60,65,66,72,78),nrow =3 ,ncol=3,byrow=TRUE)
     [,1] [,2] [,3]
[1,]   11   12   13
[2,]   55   60   65
[3,]   66   72   78
> m
     [,1] [,2] [,3]
[1,]   11   12   13
[2,]   55   60   65
[3,]   66   72   78
```

```
> #cbind-ing and rbind-ing:
> #By using cbind() and rbind() functions
> x<-c(1,2,3)
> y<-c(11,12,13)
> z<-cbind(x,y)
> z
     x  y
[1,] 1 11
[2,] 2 12
[3,] 3 13
> z<-rbind(x,y)
> z
  [,1] [,2] [,3]
x    1    2    3
y   11   12   13
> p<-3*m
> p
     [,1] [,2] [,3]
[1,]   33   36   39
[2,]  165  180  195
[3,]  198  216  234
> n<-matrix(c(4,5,6,14,15,16,24,25,26),nrow=3,ncol=3)
> #Addition of two matrices
> q<-m+n
> q
     [,1] [,2] [,3]
[1,]   15   26   37
[2,]   60   75   90
[3,]   72   88  104
> o<-matrix(c(4,5,6,14,15,16,24,25,26),nrow=3,ncol=3)
> o
     [,1] [,2] [,3]
[1,]    4   14   24
[2,]    5   15   25
[3,]    6   16   26
> #Matrix Multiplication by using %*%
> r<-m %*% o
> r
     [,1] [,2] [,3]
[1,]  182  542  902
[2,]  910 2710 4510
[3,] 1092 3252 5412
> #Transpose of Matrix
> mdash <- t(m)
> mdash
     [,1] [,2] [,3]
[1,]   11   55   66
[2,]   12   60   72
[3,]   13   65   78
> s<-matrix(c(4,5,6,14,15,16,24,25,26),nrow=3,ncol=3,byrow=TRUE)
> #Determinant of s
> s_det<-det(s)
> s_det
[1] 1.110223e-14
```

3) Factor

Source Code :-

```
#Creating a vector
x <- c("female","male","male","female")
print(x)

#Converting the vector x into a factor
#named gender
gender <- factor(x)
print(gender)

#Creating a factor with levels defined by programmer
gender <- factor(c("female","male","male","female"),
       levels=c("female","transgender","male"));

gender
#Checking for factor in R
gender <- factor(c("female","male","male","female"));
print(is.factor(gender))

#Accessing elements of a factor in R
gender <-factor(c("female","male","male","female"));
gender[3]

#More than one element can be accessed at a time.
gender<-factor(c("female","male","male","female"));
gender[c(2,4)]

#Subtract one element at a time
gender <- factor(c("female","male","male","female"));
gender[-3]

#Modification of a factor in R
gender <- factor(c("female","male","male","female"));
gender[2]<-"female"
gender
```

Output :-

```
> #Creating a vector
> x <- c("female","male","male","female")
> print(x)
[1] "female" "male"   "male"   "female"
> #Converting the vector x into a factor
> #named gender
> gender <- factor(x)
> print(gender)
[1] female male   male   female
Levels: female male
> #Creating a factor with levels defined by programmer
> gender <- factor(c("female","male","male","female"),
+           levels=c("female","transgender","male"));
> gender
[1] female male   male   female
Levels: female transgender male
> #Checking for factor in R
> gender <- factor(c("female","male","male","female"));
> print(is.factor(gender))
[1] TRUE
> #Accessing elements of a factor in R
> gender <-factor(c("female","male","male","female"));
> gender[3]
[1] male
Levels: female male
> #More than one element can be accessed at a time.
> gender<-factor(c("female","male","male","female"));
> gender[c(2,4)]
[1] male   female
Levels: female male
> #Subtract one element at a time
> gender <- factor(c("female","male","male","female"));
> gender[-3]
[1] female male   female
Levels: female male
> #Modification of a factor in R
> gender <- factor(c("female","male","male","female"));
> gender[2]<-"female"
> gender
[1] female female male   female
Levels: female male
```

4) Data Frame

Source Code :-

```
#R program to create a dataframe
#creating a data frame
friend.data <- data.frame(
  friend_id = c(1:5),
  friend_name = c("Sachin","SouraV",
                  "Dravid","Shewag",
                  "Dhoni"),
  stringsAsFactors = FALSE
)
#print the data Frame
print(friend.data)

#R program to get the structure of the data frame
#using str()
print(str(friend.data))

#R program to get the summary of the data frame
#using summary()
print(summary(friend.data))

#R program to extract data from the data frame
#Extracting friend_name column
result <- data.frame((friend.data$friend_name))
print(result)

#R program to expand the data frame
#Expanding data frame
friend.data$location <- c("kolkata","Delhi",
                          "Bangalore","Hyderabad",
                          "Chennai")
resultant <- friend.data
#Print the modified data frame
print(resultant)
#Access items using[]
friend.data[1]
#Access items using [[]]
friend.data[["friend_name"]]
#Acccess items using $
friend.data$friend_id
#Find out the number of rows and columns
dim(friend.data)


#Add rows and columns in R data frame
#Creating a dataframe representing products in a store
```

```
Products <- data.frame(
  Product_ID = c(101,102,103),
  Product_Name = c("T-Shir","Jeans","Shoes"),
  Price = c(299,399,499),
  Stock = c(50,30,25)
)

#Print the existing data frame
cat("Existing dataframe(Products): /n")
print(Products)

#Adding a new row for a new product
New_Product <- c(104,"Sunglasses",299,40)
Products <- rbind(Products,New_Product)

#Print the updated data frame after adding the new product
cat("\n Updated data frame after adding a new Product;\n")
print(Products)

#Adding a new column for 'Discount' to the data frame
Discount <- c(5,10,8,15)  #new column values for discount
Products <- cbind(Products,Discount)
#Rename the added column
colnames(Products)[ncol(Products)] <- "Discount"
#Renaming the last column


#print the updated data frame after adding the new column
cat("\n Updated data frame after adding a new column 'Disount':\n")
print(Products)

#Combing data frames in R
#Creating two sample data frames
df1 <- data.frame(
  Name = c("Alice","Bob"),
  Age = c(25,30),
  Score = c(80,75)
)
df2 <- data.frame(
  Name = c("Charlie","David"),
  Age = c(28,21),
  Score = c(75,85)
)
#Print the existing data frames
cat("Dataframe 1:\n")
print(df1)
cat("Dataframe 2:\n")
print(df2)

#Combining the data frames using rbind()
```

```
combined_df <- rbind(df1,df2)
#Print the combined data frame
cat("\n Combined Dataframe:\n")
print(combined_df)

#Combining the data frames using cbind()
combined_df <- cbind(df1,df2)
#Print the combined data frame
cat("\n Combined Dataframe:\n")
print(combined_df)
```

Output :-

```
> #R program to create a dataframe
> #creating a data frame
> friend.data <- data.frame(
+   friend_id = c(1:5),
+   friend_name = c("Sachin","SouraV",
+                   "Dravid","Shewag",
+                   "Dhoni"),
+   stringsAsFactors = FALSE
+ )
> #print the data frame
> print(friend.data)
  friend_id friend_name
1         1      Sachin
2         2      SouraV
3         3      Dravid
4         4      Shewag
5         5       Dhoni
> #R program to get the structure of the data frame
> #using str()
> print(str(friend.data))
'data.frame':   5 obs. of  2 variables:
 $ friend_id  : int  1 2 3 4 5
 $ friend_name: chr  "Sachin" "SouraV" "Dravid" "Shewag" ...
NULL
> #R program to get the summary of the data frame
> #using summary()
> print(summary(friend.data))
   friend_id   friend_name
 Min.   :1   Length:5
 1st Qu.:2   Class :character
 Median :3   Mode  :character
 Mean   :3
 3rd Qu.:4
 Max.   :5
> #R program to extract data from the data frame
> #Extracting friend_name column
> result <- data.frame((friend.data$friend_name))
> print(result)
  X.friend.data.friend_name.
1                     Sachin
2                     SouraV
3                     Dravid
4                     Shewag
5                      Dhoni
> #R program to expand the data frame
> #Expanding data frame
> friend.data$location <- c("kolkata","Delhi",
+                           "Bangalore","Hyderabad",
+                           "Chennai")
> resultant <- friend.data
> #Print the modified data frame
> print(resultant)
  friend_id friend_name  location
1         1      Sachin   kolkata
2         2      SouraV     Delhi
3         3      Dravid Bangalore
4         4      Shewag Hyderabad
5         5       Dhoni   Chennai
```

```
> #Access items using[]
> friend.data[1]
  friend_id
1         1
2         2
3         3
4         4
5         5
> #Access items using [[]]
> friend.data[["friend_name"]]
[1] "Sachin" "SouraV" "Dravid" "Shewag" "Dhoni"
> #Acccess items using $
> friend.data$friend_id
[1] 1 2 3 4 5
> #Find out the number of rows and columns
> dim(friend.data)
[1] 5 3
> #Add rows and columns in R data frame
> #Creating a dataframe representing products in a store
> Products <- data.frame(
+   Product_ID = c(101,102,103),
+   Product_Name = c("T-Shir","Jeans","Shoes"),
+   Price = c(299,399,499),
+   Stock = c(50,30,25)
+ )
> #Print the existing data frame
> cat("Existing dataframe(Products): /n")
Existing dataframe(Products): /n
> print(Products)
  Product_ID Product_Name Price Stock
1        101       T-Shir   299    50
2        102        Jeans   399    30
3        103        Shoes   499    25
> #Adding a new row for a new product
> New_Product <- c(104,"Sunglasses",299,40)
> Products <- rbind(Products,New_Product)
> #Print the updated data frame after adding the new product
> cat("\n Updated data frame after adding a new Product;\n")

 Updated data frame after adding a new Product;
> print(Products)
  Product_ID Product_Name Price Stock
1        101       T-Shir   299    50
2        102        Jeans   399    30
3        103        Shoes   499    25
4        104   Sunglasses   299    40
> #Adding a new column for 'Discount' to the data frame
> Discount <- c(5,10,8,15)  #new column values for discount
> Products <- cbind(Products,Discount)
> #Rename the added column
> colnames(Products)[ncol(Products)] <- "Discount"
> #print the updated data frame after adding the new column
> cat("\n Updated data frame after adding a new column 'Disount':\n")

 Updated data frame after adding a new column 'Disount':
> print(Products)
  Product_ID Product_Name Price Stock Discount
1        101       T-Shir   299    50        5
2        102        Jeans   399    30       10
3        103        Shoes   499    25        8
4        104   Sunglasses   299    40       15
```

```
> #Combing data frames in R
> #Creating two sample data frames
> df1 <- data.frame(
+   Name = c("Alice","Bob"),
+   Age = c(25,30),
+   Score = c(80,75)
+ )
> df2 <- data.frame(
+   Name = c("Charlie","David"),
+   Age = c(28,21),
+   Score = c(75,85)
+ )
> #Print the existing data frames
> cat("Dataframe 1:\n")
Dataframe 1:
> print(df1)
   Name Age Score
1 Alice  25    80
2   Bob  30    75
> cat("Dataframe 2:\n")
Dataframe 2:
> print(df2)
     Name Age Score
1 Charlie  28    75
2   David  21    85
> #Combining the data frames using rbind()
> combined_df <- rbind(df1,df2)
> #Print the combined data frame
> cat("\n Combined Dataframe:\n")

 Combined Dataframe:
> print(combined_df)
     Name Age Score
1   Alice  25    80
2     Bob  30    75
3 Charlie  28    75
4   David  21    85
> #Combining the data frames using cbind()
> combined_df <- cbind(df1,df2)
> #Print the combined data frame
> cat("\n Combined Dataframe:\n")

 Combined Dataframe:
> print(combined_df)
   Name Age Score    Name Age Score
1 Alice  25    80 Charlie  28    75
2   Bob  30    75   David  21    85
```