

Assignment 3

Section A

1. What is a data frame in R?

A **data frame** is a two-dimensional data structure in R that stores data in rows and columns, where each column can have a different data type.

2. Which function is used to create a data frame in R?

`data.frame()`

3. How do you check the structure of a data frame?

`str(df)`

4. Write the command to display the first six rows of a data frame.

`head(df)`

5. How do you find the number of rows in a data frame?

`nrow(df)`

Section B

1. Difference between Data Frame and Matrix

Feature	Data Frame	Matrix
Data Type	Different types allowed	Same type only
Usage	Real-world data	Mathematical data
Flexibility	More flexible	Less flexible

2. Add a New Column to a Data Frame

```
df$Age <- c(20, 21, 22)
```

OR

```
df["Age"] <- c(20, 21, 22)
```

3. Rename Columns of a Data Frame

```
colnames(df) <- c("ID", "Name", "Marks")
```

OR

```
names(df) <- c("ID", "Name", "Marks")
```

4. Extract a Specific Column

```
df$Name
```

OR

```
df[, "Name"]
```

5. Use of subset() Function

The subset() function is used to filter rows and columns based on conditions.

Example:

```
subset(df, Marks > 60)
```

This selects students with marks above 60.

Section C

1. Methods to Create a Data Frame

(a) Using data.frame()

```
df <- data.frame( RollNo = c(1,2,3), Name = c("A", "B", "C"), Marks = c(70, 80, 90) )
```

(b) From Vectors

```
roll <- c(1,2,3) name <- c("A","B","C") marks <- c(60,70,80) df <- data.frame(roll, name, marks)
```

(c) From Files

```
df <- read.csv("data.csv")
```

(d) From Lists

```
lst <- list(ID=1:3, Name=c("A","B","C")) df <- as.data.frame(lst)
```

2. Accessing Rows and Columns

(a) Using Index

```
df[1, ] # First row df[, 2] # Second column df[2,3] # Row 2, Column 3
```

(b) Using Column Name

```
df$Marks
```

(c) Using subset()

```
subset(df, Marks > 75)
```

(d) Using head() and tail()

```
head(df) tail(df)
```

3. Handling Missing Values (NA)

(a) Check Missing Values

```
is.na(df)
```

(b) Remove NA Values

```
na.omit(df)
```

(c) Replace NA with Value

```
df$Marks[is.na(df$Marks)] <- mean(df$Marks, na.rm=TRUE)
```

(d) Count NA Values

```
sum(is.na(df))
```

4. Merging Two Data Frames

Example:

```
df1 <- data.frame(ID=c(1,2,3), Name=c("A","B","C")) df2 <- data.frame(ID=c(1,2,3),  
Marks=c(70,80,90))
```

Merge by Common Column

```
merge(df1, df2, by="ID")
```

Types of Merge

Type	Parameter
Inner Join	default
Left Join	all.x=TRUE
Right Join	all.y=TRUE
Full Join	all=TRUE

Example:

```
merge(df1, df2, by="ID", all=TRUE)
```

5. Sorting and Ordering Data Frames

(a) Using order()

```
df[order(df$Marks), ]
```

Ascending order.

(b) Descending Order

```
df[order(-df$Marks), ]
```

(c) Using sort()

```
sort(df$Marks)
```

(Sorts only a column)

Section D

1. Create Student Data Frame

```
students <- data.frame( RollNo = c(1,2,3), Name = c("Rahul","Anita","Kiran"), Marks =  
c(75,85,90) ) print(students)
```

2. Add New Row

```
students <- rbind(students, data.frame(4,"Sita",88)) colnames(students) <-  
c("RollNo","Name","Marks")
```

3. Remove a Column

```
students$Marks <- NULL
```

OR

```
students <- students[, -3]
```

4. Find Mean of Numeric Column

```
mean(students$Marks)
```

OR

```
mean(students$Marks, na.rm=TRUE)
```

5. Filter Rows Based on Condition

Using subset()

```
subset(students, Marks > 80)
```

Using Indexing

```
students[students$Marks > 80, ]
```