NAME – SAARA ANAND

REG NO – 21BCE8156

SLOT – L55+L56

FDA LAB ASSIGNMENT 5-

1. **Check all the mathematical functions and show the results of each function.**

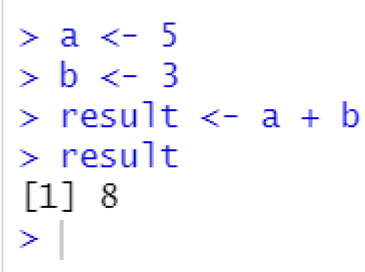
   -Addition:

   a <- 5

   b <- 3

   result <- a + b

   result



```
> a <- 5
> b <- 3
> result <- a + b
> result
[1] 8
>
```

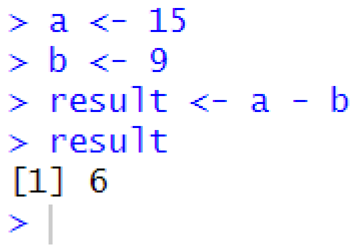
   -Subtraction:

   a <- 15

   b <- 9

   result <- a - b

   result



```
> a <- 15
> b <- 9
> result <- a - b
> result
[1] 6
> 
```

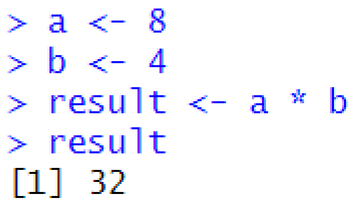
-Multiplication:

a <- 8

b <- 4

result <- a * b

result



```
> a <- 8
> b <- 4
> result <- a * b
> result
[1] 32
```

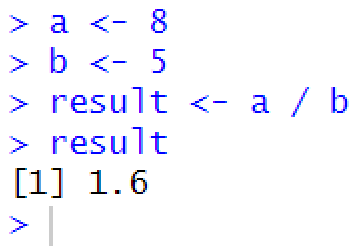
-Division:

a <- 8

b <- 5

result <- a / b

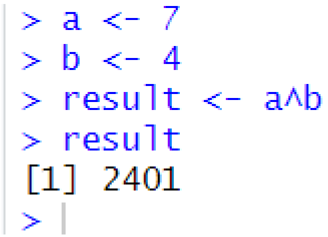
result



```
> a <- 8
> b <- 5
> result <- a / b
> result
[1] 1.6
> 
```

-Exponentiation:

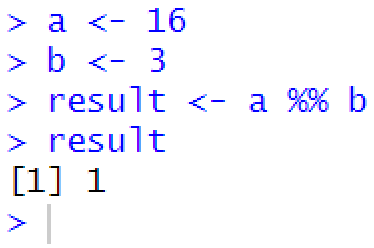
```
a <- 7

b <- 4

result <- a^b

result
```



```
> a <- 7
> b <- 4
> result <- a^b
> result
[1] 2401
> 
```

-Modulo(Remainder):

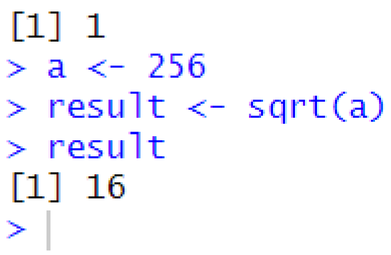
```
a <- 16

b <- 3

result <- a %% b

result
```



```
> a <- 16
> b <- 3
> result <- a %% b
> result
[1] 1
> 
```

-Square root:

```
a <- 256

result <- sqrt(a)

result
```



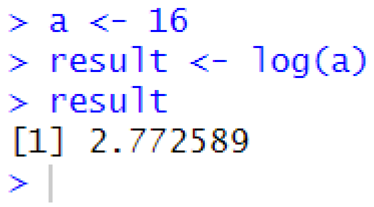
```
[1] 1
> a <- 256
> result <- sqrt(a)
> result
[1] 16
> 
```

-Logarithmic:

a <- 16

result <- log(a)

result



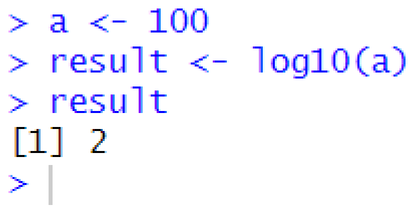
```
> a <- 16
> result <- log(a)
> result
[1] 2.772589
> 
```

-Logarithm(Base 10):

a <- 100

result <- log10(a)

result



```
> a <- 100
> result <- log10(a)
> result
[1] 2
> 
```

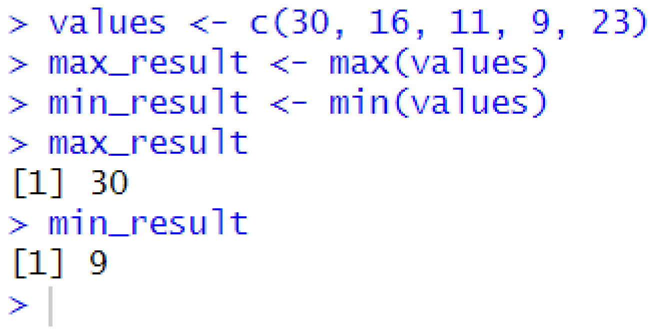
-Maximum and Minimum values:

values <- c(30, 16, 11, 9, 23)

max_result <- max(values)

min_result <- min(values)

```
max_result

min_result
```

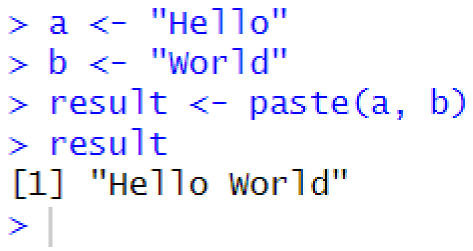


```
> values <- c(30, 16, 11, 9, 23)
> max_result <- max(values)
> min_result <- min(values)
> max_result
[1] 30
> min_result
[1] 9
> 
```

2. Implement character functions and show the results.

-Concatenation:

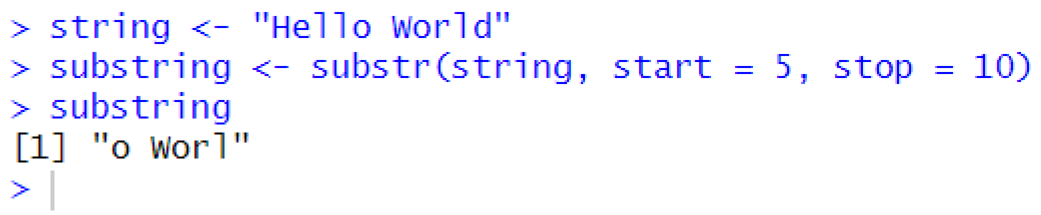
```
a <- "Hello"

b <- "World"

result <- paste(a, b)

result
```



```
> a <- "Hello"
> b <- "World"
> result <- paste(a, b)
> result
[1] "Hello World"
> 
```

-Substring Extraction:

```
string <- "Hello World"

substring <- substr(string, start = 5, stop = 10)

substring
```



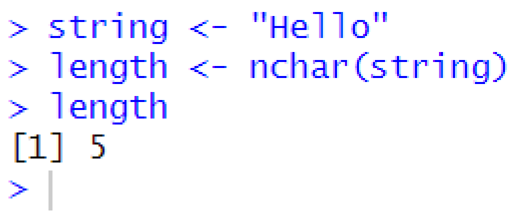
```
> string <- "Hello World"
> substring <- substr(string, start = 5, stop = 10)
> substring
[1] "o Worl"
> 
```

-Character Length:

string <- "Hello"

length <- nchar(string)

length



```
> string <- "Hello"
> length <- nchar(string)
> length
[1] 5
> 
```

-Changing Case:

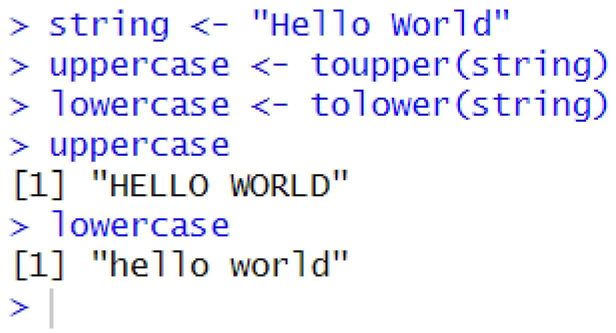
string <- "Hello World"

uppercase <- toupper(string)

lowercase <- tolower(string)

uppercase

lowercase



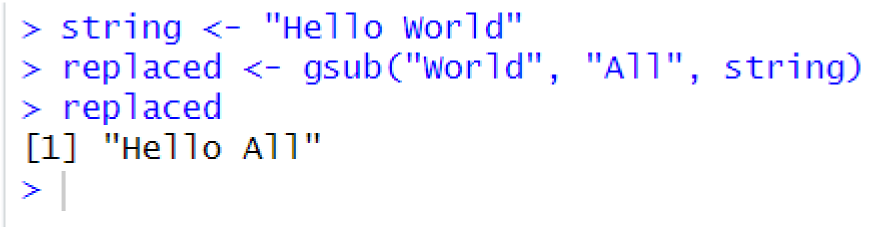
```
> string <- "Hello World"
> uppercase <- toupper(string)
> lowercase <- tolower(string)
> uppercase
[1] "HELLO WORLD"
> lowercase
[1] "hello world"
> 
```

-Pattern Matching and Replacement:

string <- "Hello World"

replaced <- gsub("World", "All", string)

replaced



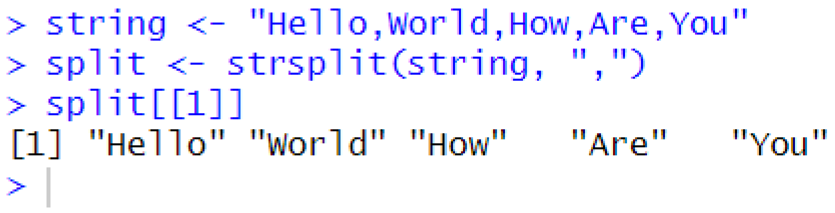
```
> string <- "Hello World"
> replaced <- gsub("World", "All", string)
> replaced
[1] "Hello All"
> 
```

-Splitting strings:

string <- "Hello,World,How,Are,You"

split <- strsplit(string, ",")

split[[1]]



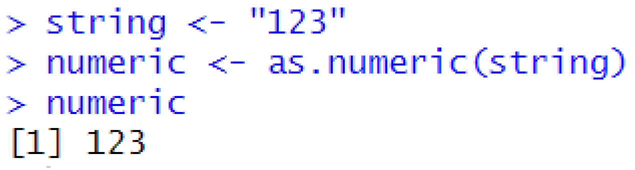
```
> string <- "Hello,World,How,Are,You"
> split <- strsplit(string, ",")
> split[[1]]
[1] "Hello" "World" "How"   "Are"   "You"  
> 
```

-Conversion to numeric:

string <- "123"

numeric <- as.numeric(string)

numeric



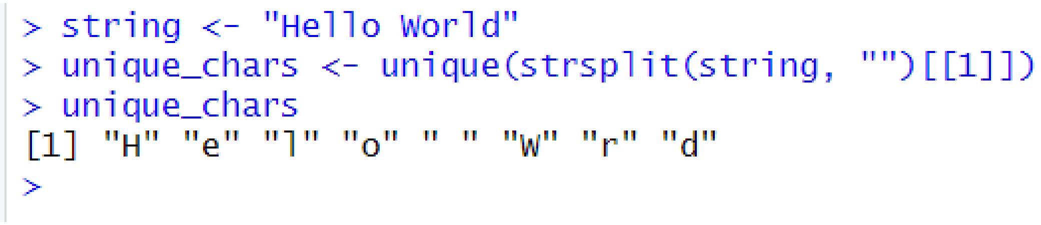
```
> string <- "123"
> numeric <- as.numeric(string)
> numeric
[1] 123
```

-Extracting unique characters:

string <- "Hello World"

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

unique_chars



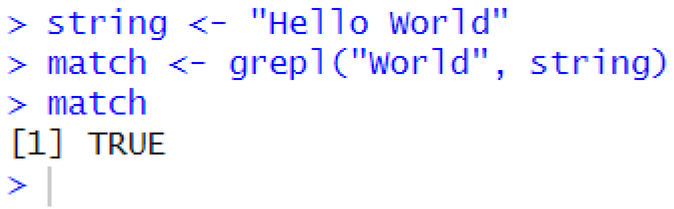
```
> string <- "Hello World"
> unique_chars <- unique(strsplit(string, "")[[1]])
> unique_chars
[1] "H" "e" "l" "o" " " "W" "r" "d"
>
```

-String Matching:

string <- "Hello World"

match <- grepl("World", string)

match



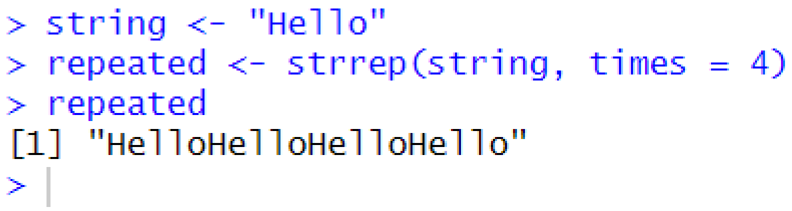
```
> string <- "Hello World"
> match <- grepl("World", string)
> match
[1] TRUE
>
```

-String Repeating:

string <- "Hello"

repeated <- strrep(string, times = 4)

repeated

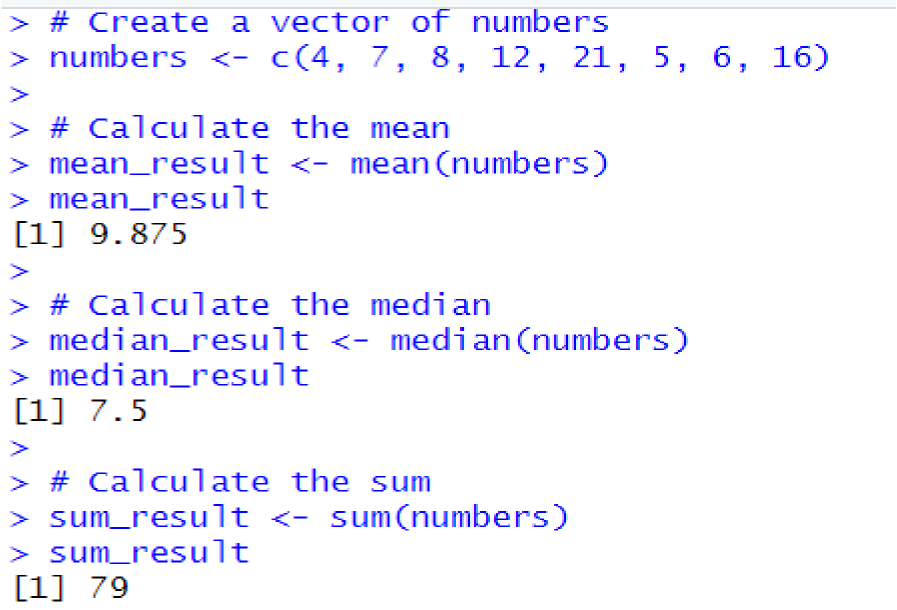


```
> string <- "Hello"
> repeated <- strrep(string, times = 4)
> repeated
[1] "HelloHelloHelloHello"
>
```

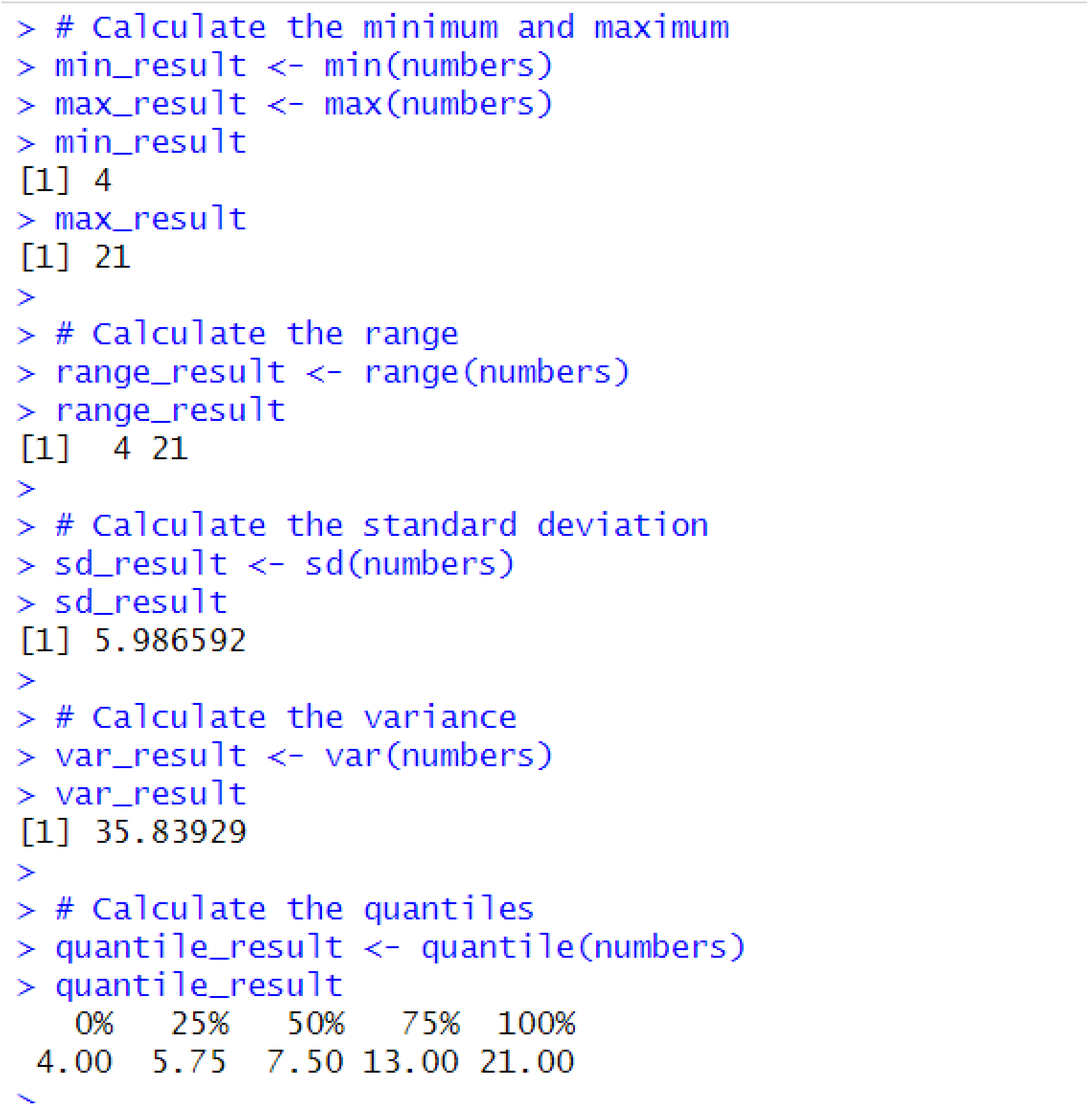
3.Create a vector of numbers and work with all statistical functions and report the results.

```
# Create a vector of numbers
numbers <- c(4, 7, 8, 12, 21, 5, 6, 16)
# Calculate the mean
mean_result <- mean(numbers)
mean_result
# Calculate the median
median_result <- median(numbers)
median_result
# Calculate the sum
sum_result <- sum(numbers)
sum_result
# Calculate the minimum and maximum
min_result <- min(numbers)
max_result <- max(numbers)
min_result
max_result
# Calculate the range
range_result <- range(numbers)
range_result
# Calculate the standard deviation
sd_result <- sd(numbers)
```

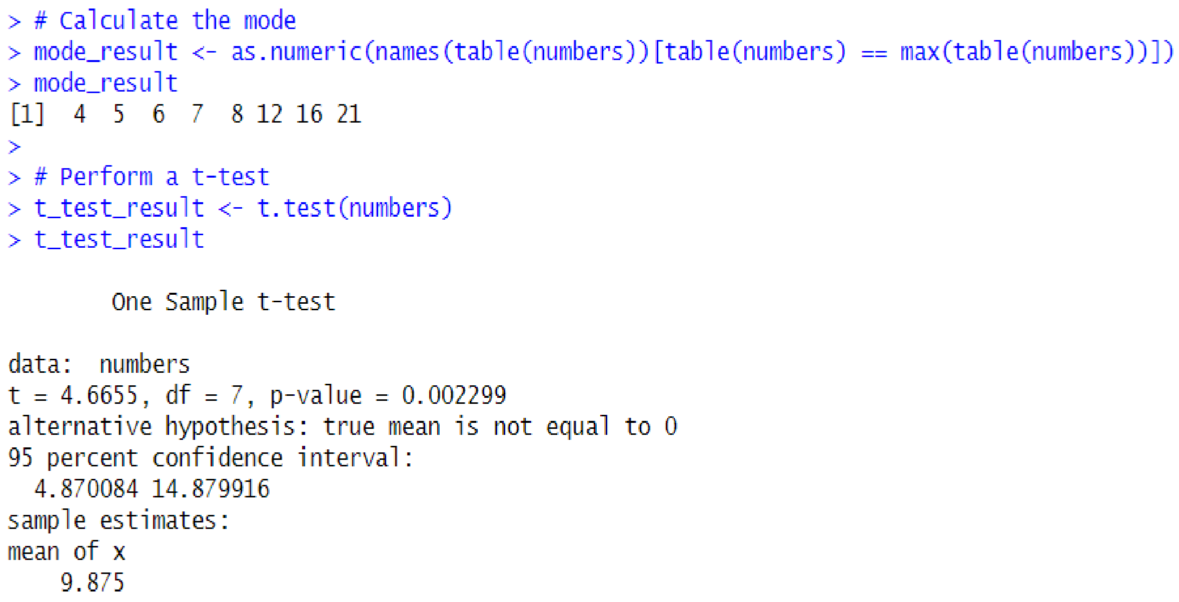
```
sd_result

# Calculate the variance

var_result <- var(numbers)

var_result

# Calculate the quantiles

quantile_result <- quantile(numbers)

quantile_result

# Calculate the mode

mode_result <- as.numeric(names(table(numbers))[table(numbers) == max(table(numbers))])

mode_result

# Perform a t-test

t_test_result <- t.test(numbers)

t_test_result
```



```
> # Create a vector of numbers
> numbers <- c(4, 7, 8, 12, 21, 5, 6, 16)
>
> # Calculate the mean
> mean_result <- mean(numbers)
> mean_result
[1] 9.875
>
> # Calculate the median
> median_result <- median(numbers)
> median_result
[1] 7.5
>
> # Calculate the sum
> sum_result <- sum(numbers)
> sum_result
[1] 79
```



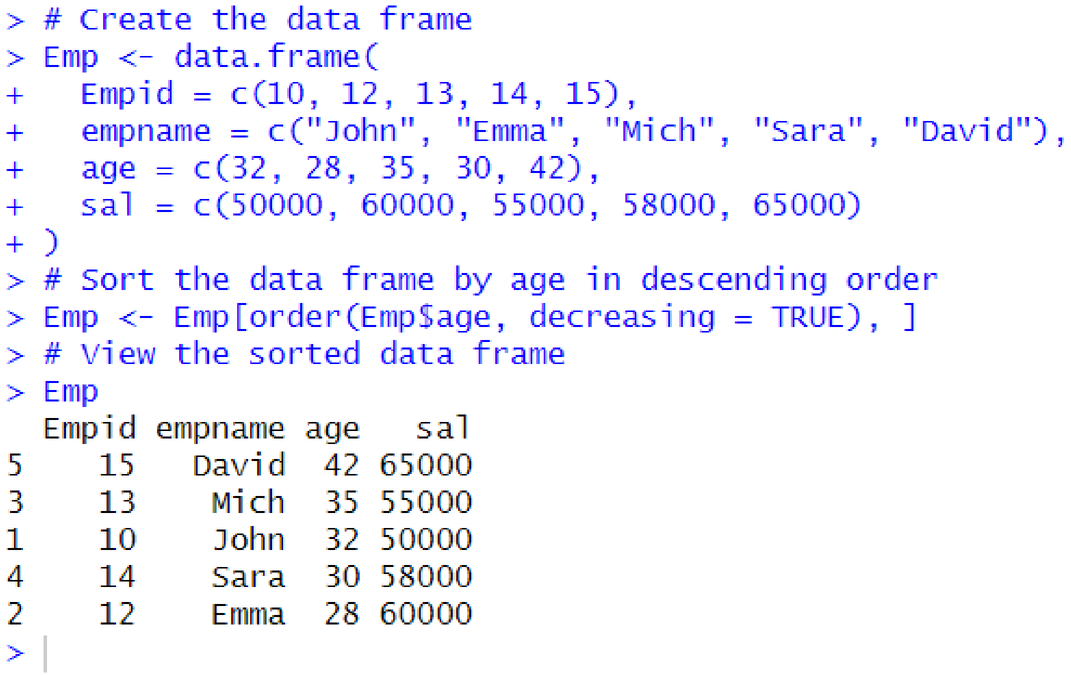
```
> # Calculate the minimum and maximum
> min_result <- min(numbers)
> max_result <- max(numbers)
> min_result
[1] 4
> max_result
[1] 21
> 
> # Calculate the range
> range_result <- range(numbers)
> range_result
[1]  4 21
> 
> # Calculate the standard deviation
> sd_result <- sd(numbers)
> sd_result
[1] 5.986592
> 
> # Calculate the variance
> var_result <- var(numbers)
> var_result
[1] 35.83929
> 
> # Calculate the quantiles
> quantile_result <- quantile(numbers)
> quantile_result
   0%   25%   50%   75%  100% 
 4.00  5.75  7.50 13.00 21.00 
>
```



```
> # Calculate the mode
> mode_result <- as.numeric(names(table(numbers))[table(numbers) == max(table(numbers))])
> mode_result
[1]  4  5  6  7  8 12 16 21
> 
> # Perform a t-test
> t_test_result <- t.test(numbers)
> t_test_result

	One Sample t-test

data:  numbers
t = 4.6655, df = 7, p-value = 0.002299
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
  4.870084 14.879916
sample estimates:
mean of x 
    9.875 
```

4.Create a data frame Emp to store (Empid, empname, age,sal).Sort the data in the descending order of age.

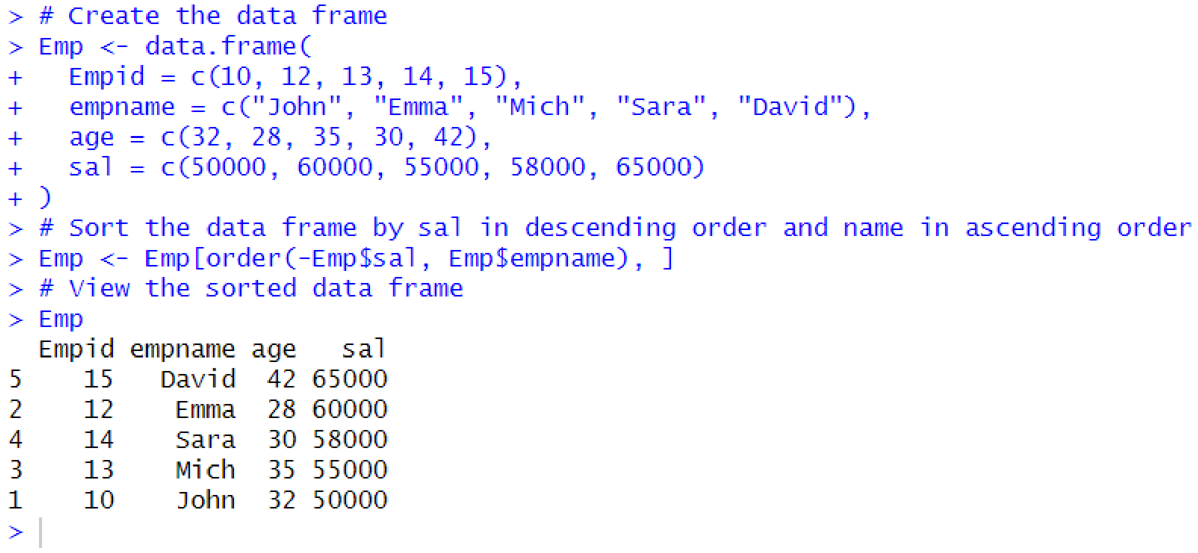
```
# Create the data frame

Emp <- data.frame(

  Empid = c(10, 12, 13, 14, 15),

  empname = c("John", "Emma", "Mich", "Sara", "David"),

  age = c(32, 28, 35, 30, 42),

  sal = c(50000, 60000, 55000, 58000, 65000)

)

# Sort the data frame by age in descending order

Emp <- Emp[order(Emp$age, decreasing = TRUE), ]

# View the sorted data frame

Emp
```



```
> # Create the data frame
> Emp <- data.frame(
+   Empid = c(10, 12, 13, 14, 15),
+   empname = c("John", "Emma", "Mich", "Sara", "David"),
+   age = c(32, 28, 35, 30, 42),
+   sal = c(50000, 60000, 55000, 58000, 65000)
+ )
> # Sort the data frame by age in descending order
> Emp <- Emp[order(Emp$age, decreasing = TRUE), ]
> # View the sorted data frame
> Emp
  Empid empname age   sal
5    15   David  42 65000
3    13    Mich  35 55000
1    10    John  32 50000
4    14    Sara  30 58000
2    12    Emma  28 60000
> 
```

5.Sort the data in Emp in the descending order of sal and ascending order of name.

```
# Create the data frame

Emp <- data.frame(

  Empid = c(10, 12, 13, 14, 15),

  empname = c("John", "Emma", "Mich", "Sara", "David"),

  age = c(32, 28, 35, 30, 42),

  sal = c(50000, 60000, 55000, 58000, 65000)

)

# Sort the data frame by sal in descending order and name in ascending order

Emp <- Emp[order(-Emp$sal, Emp$empname), ]

# View the sorted data frame

Emp
```



```
> # Create the data frame
> Emp <- data.frame(
+   Empid = c(10, 12, 13, 14, 15),
+   empname = c("John", "Emma", "Mich", "Sara", "David"),
+   age = c(32, 28, 35, 30, 42),
+   sal = c(50000, 60000, 55000, 58000, 65000)
+ )
> # Sort the data frame by sal in descending order and name in ascending order
> Emp <- Emp[order(-Emp$sal, Emp$empname), ]
> # View the sorted data frame
> Emp
  Empid empname age   sal
5    15   David  42 65000
2    12    Emma  28 60000
4    14    Sara  30 58000
3    13    Mich  35 55000
1    10    John  32 50000
> 
```

6. Add a column DeptNo to Emp.

Create the data frame

Emp <- data.frame(

  Empid = c(10, 12, 13, 14, 15),

  empname = c("John", "Emma", "Mich", "Sara", "David"),

  age = c(32, 28, 35, 30, 42),

  sal = c(50000, 60000, 55000, 58000, 65000)

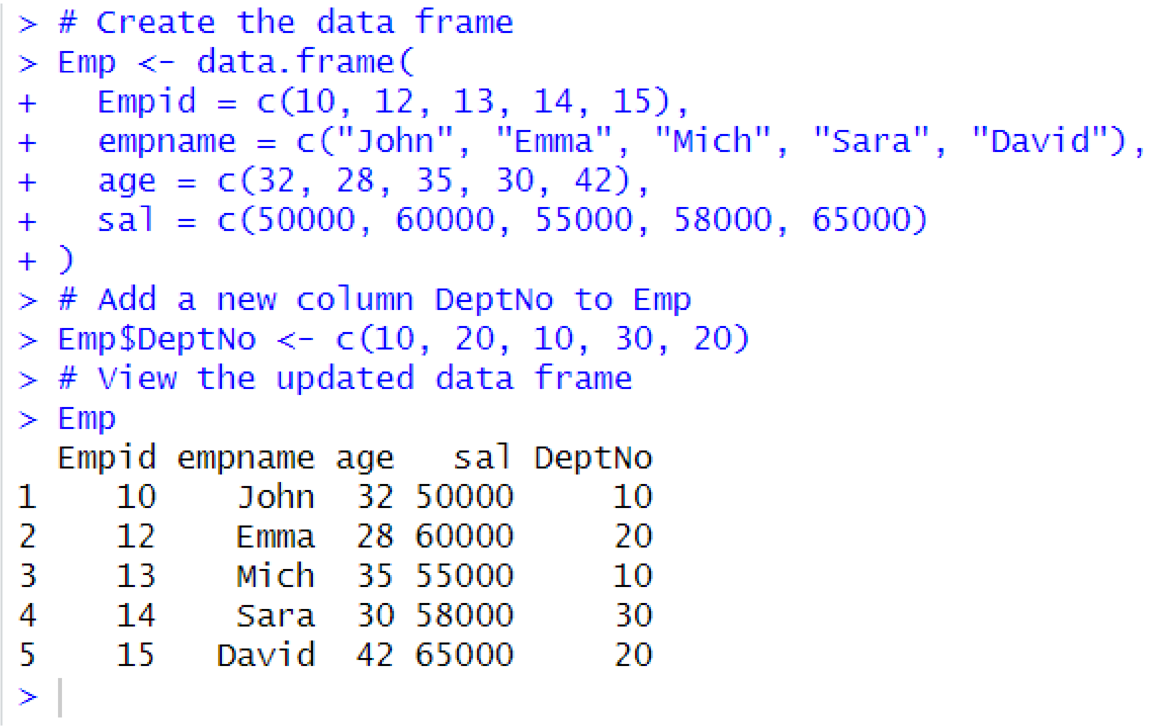
)

Add a new column DeptNo to Emp

Emp$DeptNo <- c(10, 20, 10, 30, 20)

View the updated data frame

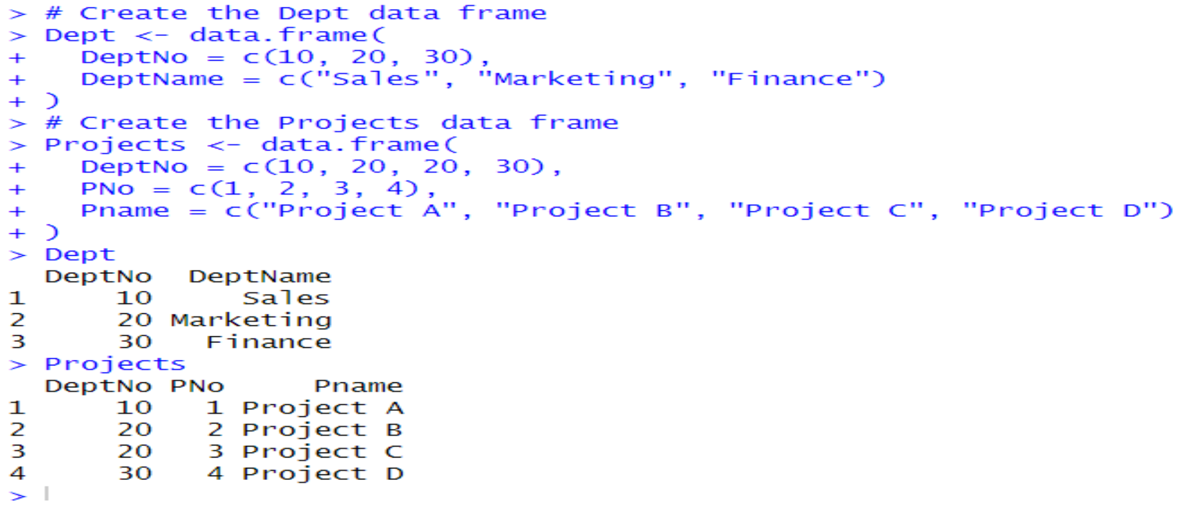
Emp



```
> # Create the data frame
> Emp <- data.frame(
+   Empid = c(10, 12, 13, 14, 15),
+   empname = c("John", "Emma", "Mich", "Sara", "David"),
+   age = c(32, 28, 35, 30, 42),
+   sal = c(50000, 60000, 55000, 58000, 65000)
+ )
> # Add a new column DeptNo to Emp
> Emp$DeptNo <- c(10, 20, 10, 30, 20)
> # View the updated data frame
> Emp
  Empid empname age   sal DeptNo
1    10    John  32 50000     10
2    12    Emma  28 60000     20
3    13    Mich  35 55000     10
4    14    Sara  30 58000     30
5    15   David  42 65000     20
> 
```

7. Create a data frame Dept(DeptNo ,DeptName) and Projects(DeptNo,PNo,Pname).

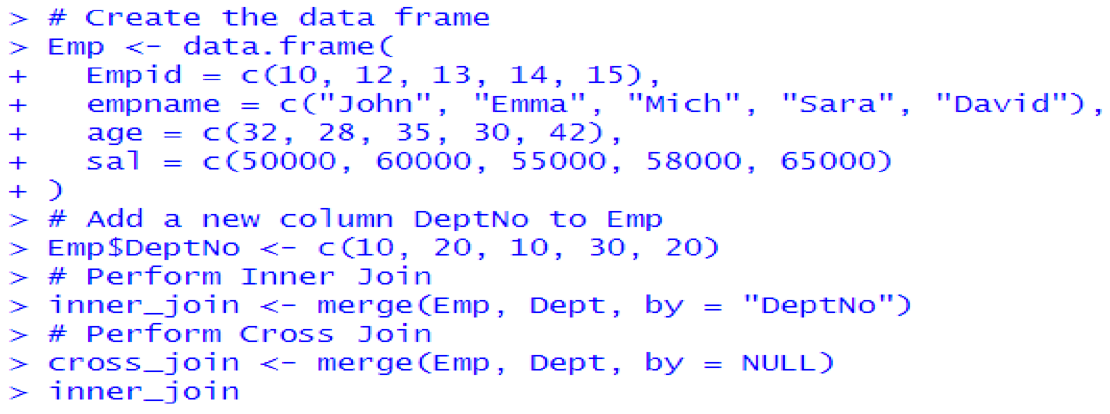
```
# Create the Dept data frame

Dept <- data.frame(

  DeptNo = c(10, 20, 30),

  DeptName = c("Sales", "Marketing", "Finance")

)

# Create the Projects data frame

Projects <- data.frame(

  DeptNo = c(10, 20, 20, 30),

  PNo = c(1, 2, 3, 4),

  Pname = c("Project A", "Project B", "Project C", "Project D")

)

Dept

Projects
```



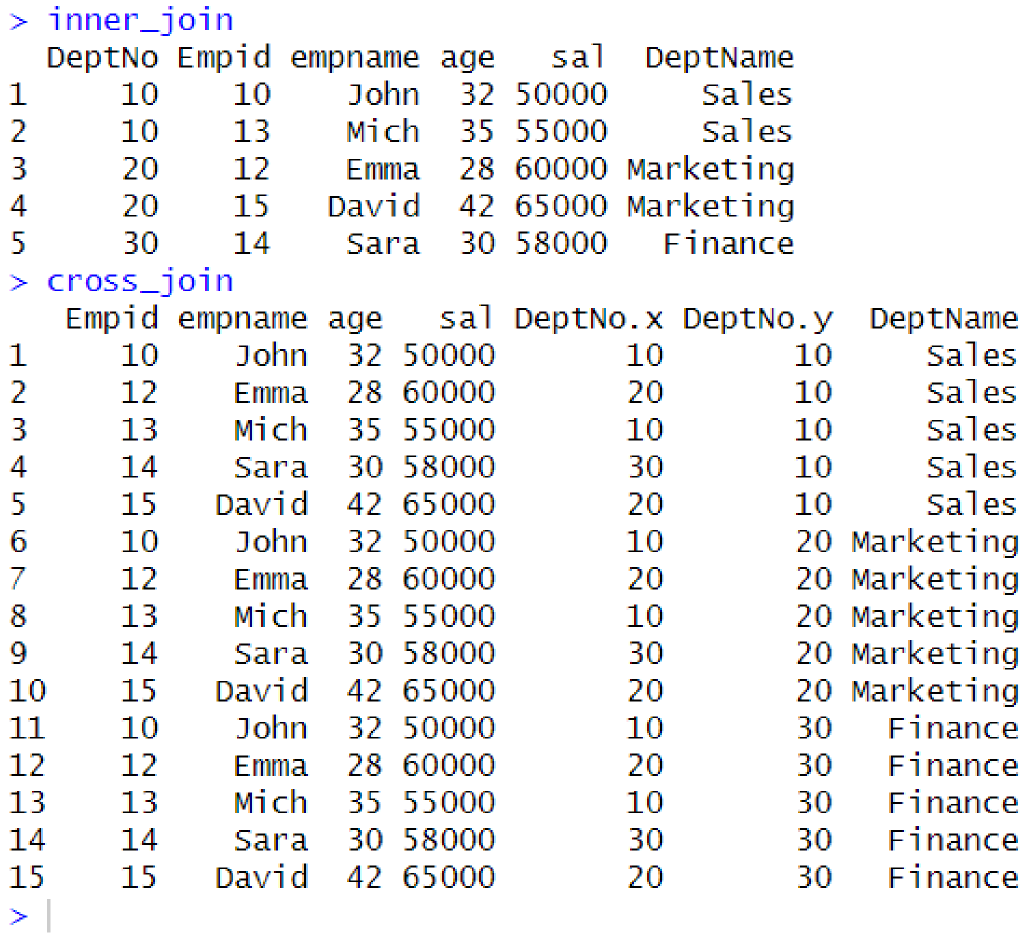
```
> # Create the Dept data frame
> Dept <- data.frame(
+   DeptNo = c(10, 20, 30),
+   DeptName = c("Sales", "Marketing", "Finance")
+ )
> # Create the Projects data frame
> Projects <- data.frame(
+   DeptNo = c(10, 20, 20, 30),
+   PNo = c(1, 2, 3, 4),
+   Pname = c("Project A", "Project B", "Project C", "Project D")
+ )
> Dept
  DeptNo  DeptName
1     10     Sales
2     20 Marketing
3     30   Finance
> Projects
  DeptNo PNo     Pname
1     10   1 Project A
2     20   2 Project B
3     20   3 Project C
4     30   4 Project D
> 
```

8. Perform Inner Join and Cross Join using Emp and Dept.

```
# Create the data frame

Emp <- data.frame(

  Empid = c(10, 12, 13, 14, 15),

  empname = c("John", "Emma", "Mich", "Sara", "David"),

  age = c(32, 28, 35, 30, 42),

  sal = c(50000, 60000, 55000, 58000, 65000)

)

# Add a new column DeptNo to Emp

Emp$DeptNo <- c(10, 20, 10, 30, 20)

# Perform Inner Join

inner_join <- merge(Emp, Dept, by = "DeptNo")

# Perform Cross Join

cross_join <- merge(Emp, Dept, by = NULL)

inner_join

cross_join
```



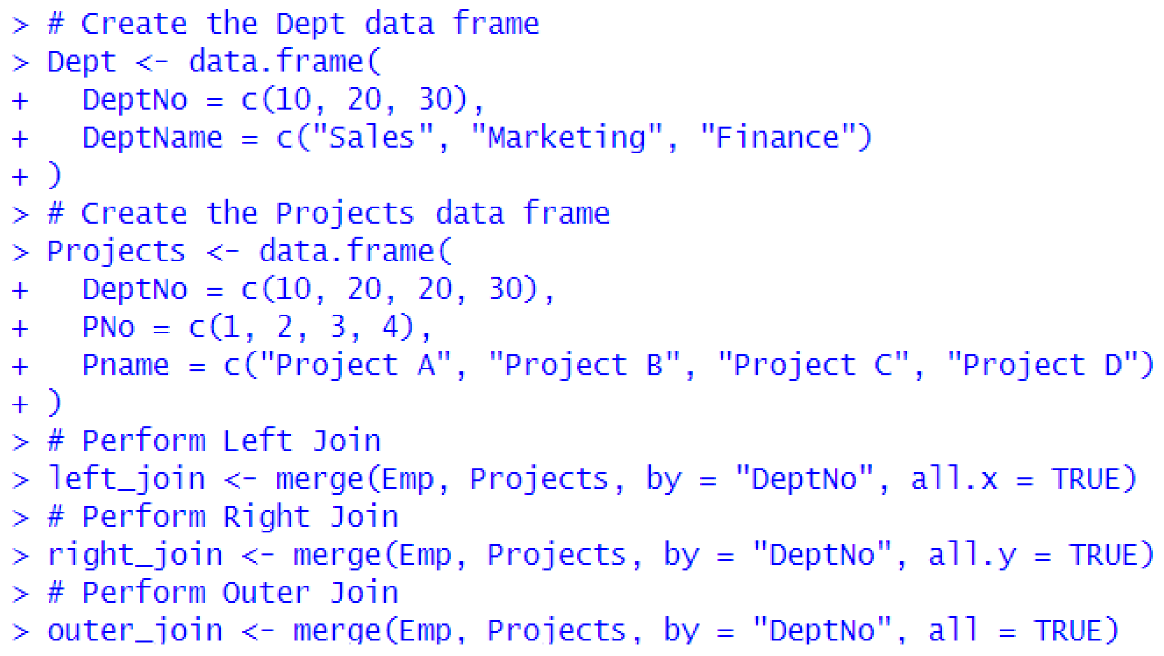
```
> # Create the data frame
> Emp <- data.frame(
+   Empid = c(10, 12, 13, 14, 15),
+   empname = c("John", "Emma", "Mich", "Sara", "David"),
+   age = c(32, 28, 35, 30, 42),
+   sal = c(50000, 60000, 55000, 58000, 65000)
+ )
> # Add a new column DeptNo to Emp
> Emp$DeptNo <- c(10, 20, 10, 30, 20)
> # Perform Inner Join
> inner_join <- merge(Emp, Dept, by = "DeptNo")
> # Perform Cross Join
> cross_join <- merge(Emp, Dept, by = NULL)
> inner_join
```



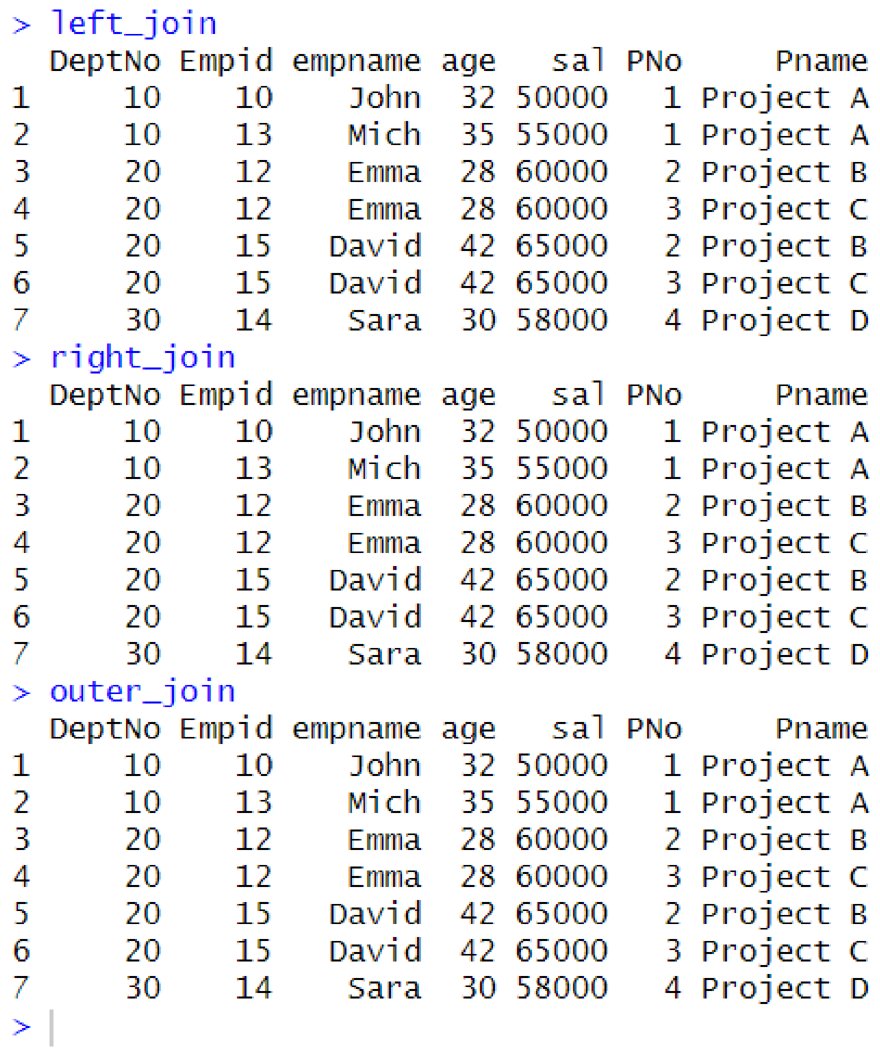
```
> inner_join
  DeptNo Empid empname age   sal  DeptName
1     10    10    John  32 50000     Sales
2     10    13    Mich  35 55000     Sales
3     20    12    Emma  28 60000 Marketing
4     20    15   David  42 65000 Marketing
5     30    14    Sara  30 58000   Finance
> cross_join
   Empid empname age   sal DeptNo.x DeptNo.y  DeptName
1     10    John  32 50000       10       10     Sales
2     12    Emma  28 60000       20       10     Sales
3     13    Mich  35 55000       10       10     Sales
4     14    Sara  30 58000       30       10     Sales
5     15   David  42 65000       20       10     Sales
6     10    John  32 50000       10       20 Marketing
7     12    Emma  28 60000       20       20 Marketing
8     13    Mich  35 55000       10       20 Marketing
9     14    Sara  30 58000       30       20 Marketing
10    15   David  42 65000       20       20 Marketing
11    10    John  32 50000       10       30   Finance
12    12    Emma  28 60000       20       30   Finance
13    13    Mich  35 55000       10       30   Finance
14    14    Sara  30 58000       30       30   Finance
15    15   David  42 65000       20       30   Finance
> 
```

9.Perform Left Join, Right Join, Outer Join using Emp and Project.

```
# Create the Dept data frame

Dept <- data.frame(

  DeptNo = c(10, 20, 30),

  DeptName = c("Sales", "Marketing", "Finance")

)

# Create the Projects data frame

Projects <- data.frame(

  DeptNo = c(10, 20, 20, 30),

  PNo = c(1, 2, 3, 4),
  Pname = c("Project A", "Project B", "Project C", "Project D")

)

# Perform Left Join

left_join <- merge(Emp, Projects, by = "DeptNo", all.x = TRUE)

# Perform Right Join

right_join <- merge(Emp, Projects, by = "DeptNo", all.y = TRUE)

# Perform Outer Join

outer_join <- merge(Emp, Projects, by = "DeptNo", all = TRUE)

left_join

right_join

outer_join
```



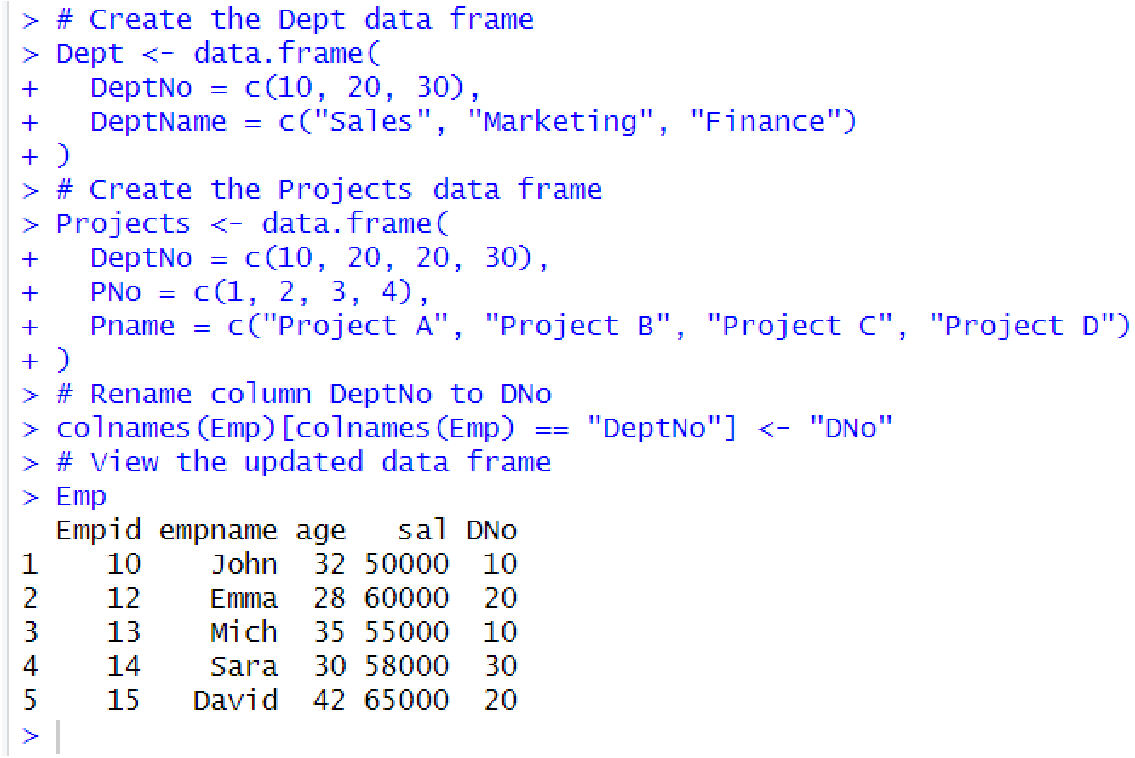
```
> # Create the Dept data frame
> Dept <- data.frame(
+   DeptNo = c(10, 20, 30),
+   DeptName = c("Sales", "Marketing", "Finance")
+ )
> # Create the Projects data frame
> Projects <- data.frame(
+   DeptNo = c(10, 20, 20, 30),
+   PNo = c(1, 2, 3, 4),
+   Pname = c("Project A", "Project B", "Project C", "Project D")
+ )
> # Perform Left Join
> left_join <- merge(Emp, Projects, by = "DeptNo", all.x = TRUE)
> # Perform Right Join
> right_join <- merge(Emp, Projects, by = "DeptNo", all.y = TRUE)
> # Perform Outer Join
> outer_join <- merge(Emp, Projects, by = "DeptNo", all = TRUE)
```



```
> left_join
  DeptNo Empid empname age   sal PNo     Pname
1     10    10    John  32 50000   1 Project A
2     10    13    Mich  35 55000   1 Project A
3     20    12    Emma  28 60000   2 Project B
4     20    12    Emma  28 60000   3 Project C
5     20    15   David  42 65000   2 Project B
6     20    15   David  42 65000   3 Project C
7     30    14    Sara  30 58000   4 Project D
> right_join
  DeptNo Empid empname age   sal PNo     Pname
1     10    10    John  32 50000   1 Project A
2     10    13    Mich  35 55000   1 Project A
3     20    12    Emma  28 60000   2 Project B
4     20    12    Emma  28 60000   3 Project C
5     20    15   David  42 65000   2 Project B
6     20    15   David  42 65000   3 Project C
7     30    14    Sara  30 58000   4 Project D
> outer_join
  DeptNo Empid empname age   sal PNo     Pname
1     10    10    John  32 50000   1 Project A
2     10    13    Mich  35 55000   1 Project A
3     20    12    Emma  28 60000   2 Project B
4     20    12    Emma  28 60000   3 Project C
5     20    15   David  42 65000   2 Project B
6     20    15   David  42 65000   3 Project C
7     30    14    Sara  30 58000   4 Project D
> 
```

10.Rename Column DeptNo to DNo in Emp.

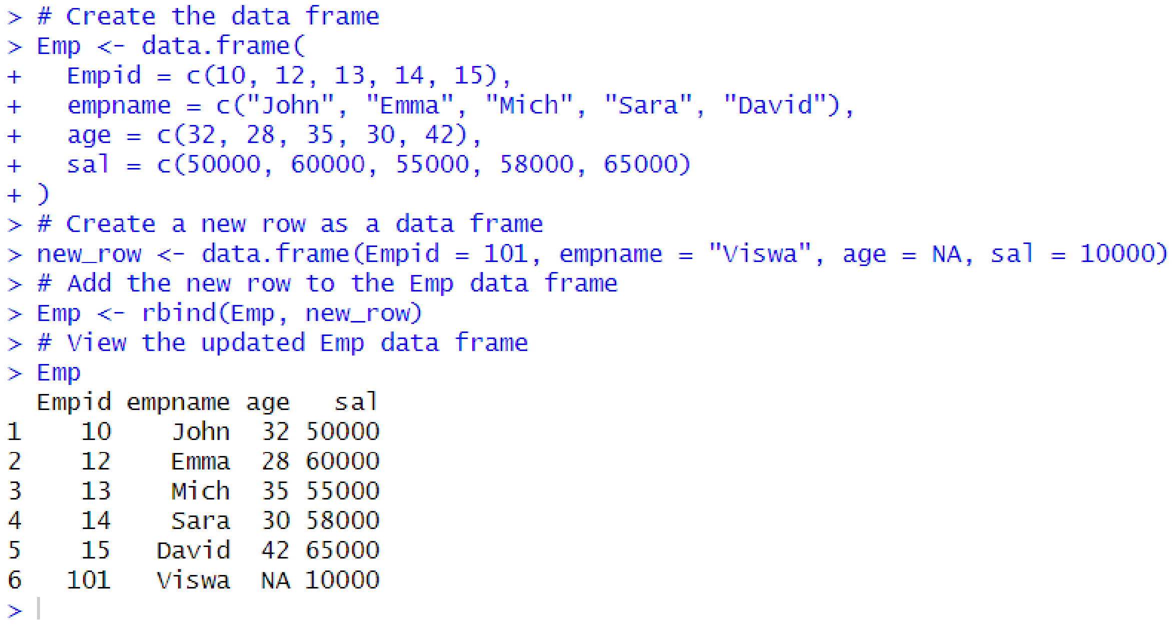
```
# Create the Dept data frame

Dept <- data.frame(

  DeptNo = c(10, 20, 30),

  DeptName = c("Sales", "Marketing", "Finance")

)

# Create the Projects data frame

Projects <- data.frame(

  DeptNo = c(10, 20, 20, 30),
  PNo = c(1, 2, 3, 4),

  Pname = c("Project A", "Project B", "Project C", "Project D")

)

# Rename column DeptNo to DNo

colnames(Emp)[colnames(Emp) == "DeptNo"] <- "DNo"

# View the updated data frame

Emp
```



```
> # Create the Dept data frame
> Dept <- data.frame(
+   DeptNo = c(10, 20, 30),
+   DeptName = c("Sales", "Marketing", "Finance")
+ )
> # Create the Projects data frame
> Projects <- data.frame(
+   DeptNo = c(10, 20, 20, 30),
+   PNo = c(1, 2, 3, 4),
+   Pname = c("Project A", "Project B", "Project C", "Project D")
+ )
> # Rename column DeptNo to DNo
> colnames(Emp)[colnames(Emp) == "DeptNo"] <- "DNo"
> # View the updated data frame
> Emp
  Empid empname age   sal DNo
1    10    John  32 50000  10
2    12    Emma  28 60000  20
3    13    Mich  35 55000  10
4    14    Sara  30 58000  30
5    15   David  42 65000  20
> 
```

11. Add a new Emp (101,“Viswa”, NA, 10000).

```
# Create the data frame

Emp <- data.frame(

  Empid = c(10, 12, 13, 14, 15),

  empname = c("John", "Emma", "Mich", "Sara", "David"),
  age = c(32, 28, 35, 30, 42),

  sal = c(50000, 60000, 55000, 58000, 65000)

)

# Create a new row as a data frame

new_row <- data.frame(Empid = 101, empname = "Viswa", age = NA, sal = 10000)

# Add the new row to the Emp data frame

Emp <- rbind(Emp, new_row)

# View the updated Emp data frame

Emp
```



```
> # Create the data frame
> Emp <- data.frame(
+   Empid = c(10, 12, 13, 14, 15),
+   empname = c("John", "Emma", "Mich", "Sara", "David"),
+   age = c(32, 28, 35, 30, 42),
+   sal = c(50000, 60000, 55000, 58000, 65000)
+ )
> # Create a new row as a data frame
> new_row <- data.frame(Empid = 101, empname = "Viswa", age = NA, sal = 10000)
> # Add the new row to the Emp data frame
> Emp <- rbind(Emp, new_row)
> # View the updated Emp data frame
> Emp
  Empid empname age   sal
1    10    John  32 50000
2    12    Emma  28 60000
3    13    Mich  35 55000
4    14    Sara  30 58000
5    15   David  42 65000
6   101   Viswa  NA 10000
> 
```

12.Replace missing value of age with mean, median and a value.

```
# Calculate the mean and median of the age column

mean_age <- mean(Emp$age, na.rm = TRUE)

median_age <- median(Emp$age, na.rm = TRUE)
```

Replace missing values with mean, median, and a specific value (40)

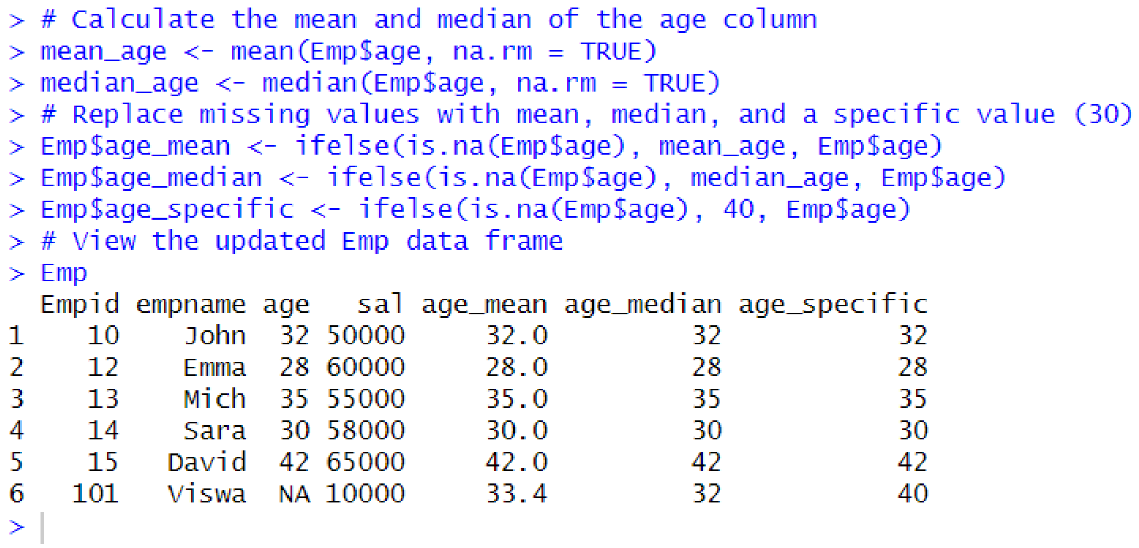
Emp$age_mean <- ifelse(is.na(Emp$age), mean_age, Emp$age)

Emp$age_median <- ifelse(is.na(Emp$age), median_age, Emp$age)

Emp$age_specific <- ifelse(is.na(Emp$age), 40, Emp$age)

View the updated Emp data frame

Emp

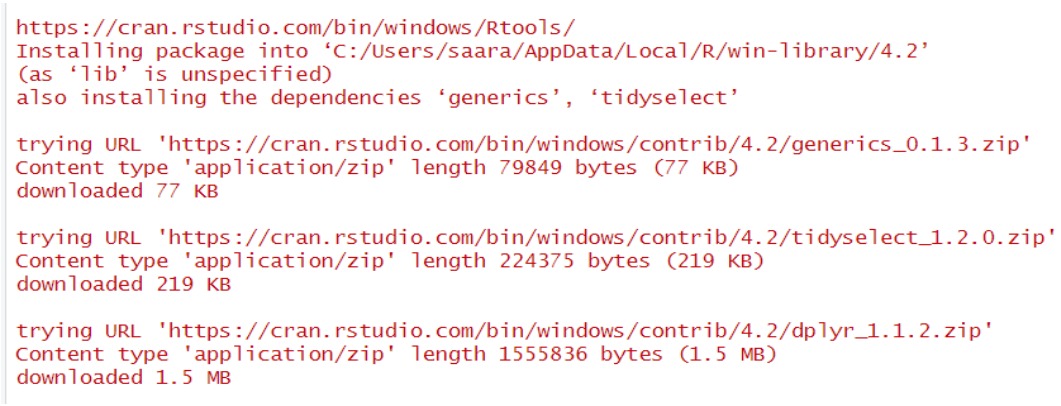


```
> # Calculate the mean and median of the age column
> mean_age <- mean(Emp$age, na.rm = TRUE)
> median_age <- median(Emp$age, na.rm = TRUE)
> # Replace missing values with mean, median, and a specific value (30)
> Emp$age_mean <- ifelse(is.na(Emp$age), mean_age, Emp$age)
> Emp$age_median <- ifelse(is.na(Emp$age), median_age, Emp$age)
> Emp$age_specific <- ifelse(is.na(Emp$age), 40, Emp$age)
> # View the updated Emp data frame
> Emp
  Empid empname age   sal age_mean age_median age_specific
1    10    John  32 50000     32.0         32           32
2    12    Emma  28 60000     28.0         28           28
3    13    Mich  35 55000     35.0         35           35
4    14    Sara  30 58000     30.0         30           30
5    15   David  42 65000     42.0         42           42
6   101   Viswa  NA 10000     33.4         32           40
> 
```

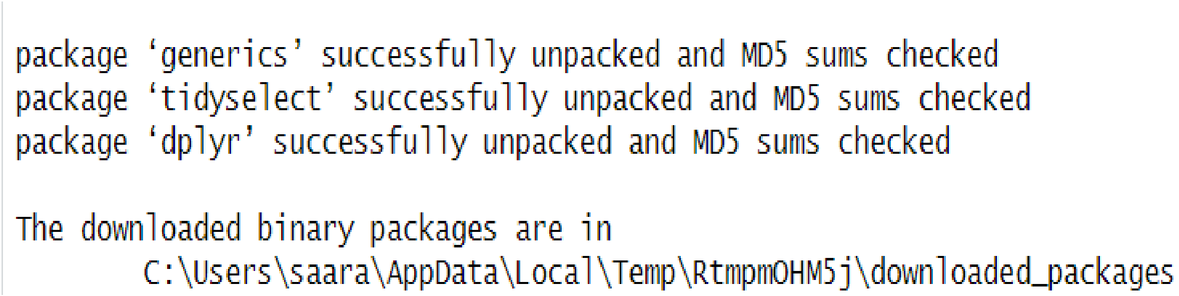
13. Installation of dplyr package and its operations.

# Install dplyr package

install.packages("dplyr")

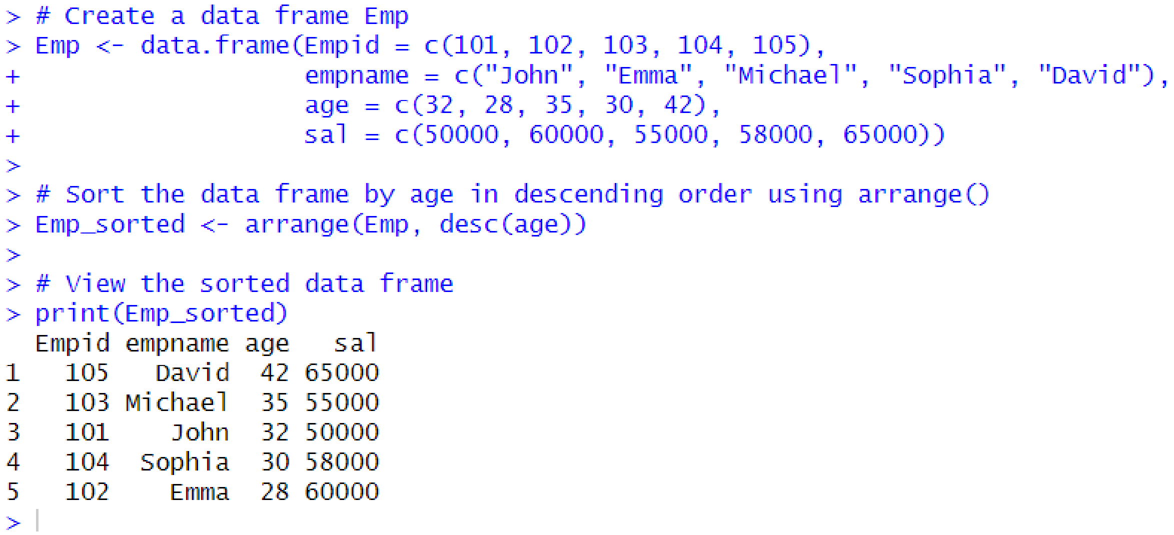


```
https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/saara/AppData/Local/R/win-library/4.2’
(as ‘lib’ is unspecified)
also installing the dependencies ‘generics’, ‘tidyselect’

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/generics_0.1.3.zip'
Content type 'application/zip' length 79849 bytes (77 KB)
downloaded 77 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/tidyselect_1.2.0.zip'
Content type 'application/zip' length 224375 bytes (219 KB)
downloaded 219 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/dplyr_1.1.2.zip'
Content type 'application/zip' length 1555836 bytes (1.5 MB)
downloaded 1.5 MB
```



```
package ‘generics’ successfully unpacked and MD5 sums checked
package ‘tidyselect’ successfully unpacked and MD5 sums checked
package ‘dplyr’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\saara\AppData\Local\Temp\RtmpmOHM5j\downloaded_packages
```

```
library(dplyr)

# Create a data frame Emp

Emp <- data.frame(Empid = c(101, 102, 103, 104, 105),

                  empname = c("John", "Emma", "Michael", "Sophia", "David"),

                  age = c(32, 28, 35, 30, 42),

                  sal = c(50000, 60000, 55000, 58000, 65000))


# Sort the data frame by age in descending order using arrange()

Emp_sorted <- arrange(Emp, desc(age))


# View the sorted data frame

print(Emp_sorted)
```



```
> # Create a data frame Emp
> Emp <- data.frame(Empid = c(101, 102, 103, 104, 105),
+                   empname = c("John", "Emma", "Michael", "Sophia", "David"),
+                   age = c(32, 28, 35, 30, 42),
+                   sal = c(50000, 60000, 55000, 58000, 65000))
>
> # Sort the data frame by age in descending order using arrange()
> Emp_sorted <- arrange(Emp, desc(age))
>
> # View the sorted data frame
> print(Emp_sorted)
  Empid empname age   sal
1   105   David  42 65000
2   103 Michael  35 55000
3   101    John  32 50000
4   104  Sophia  30 58000
5   102    Emma  28 60000
> 
```

--X--

Thank you!