**Ex.No**.3 **DPLYR:DATA MANIPULATION**

**Date**: 1-08-23

**Aim**

To implement the Matrices in R programming in the experiments and learn about them.

**Procedure**

1. To do programming in R, first install “RStudio” and “R” in the system. RStudio is an integrated development environment [IDE] for R and python.
2. Select the File in taskbar →open New file →R script or use shortcut “ctrl+shift+N”
3. Install the ‘dplyr’ package and load it in R.
4. Write the program in the script and save it using the extension R.
5. Run the program by clicking Run option or use the shortcut “ctrl+enter”.
6. See the output in the console tab.

**Concepts Applied**

* Data manipulation using ‘dplyr’ package.

**DPLYR PACKAGE**

The dplyr package provides several important functions for data manipulation. These are: -

1. **filter ( ) function:** for choosing cases and using their values as base for doing so. It can filter out the prompt user provided.

**Script**

#filter() function

#Create a data frame with missing data

d <- data.frame( name = c("Abhi", "Bhavesh",

"Chaman", "Dimri"),

age = c(7, 5, 9, 16),

ht = c(46, NA, NA, 69),

school = c("yes", "yes", "no", "no") )

# Finding rows with NA value

d %>% filter(is.na(ht))

# Finding rows with no NA value

d %>% filter(!is.na(ht))

**Output**

name age ht school

1 Bhavesh 5 NA yes

2 Chaman 9 NA no

name age ht school

1 Abhi 7 46 yes

2 Dimri 16 69 no

1. **arrange ( ) function**

For reordering of the cases, the arrange ( ) function can be used.

**Script**

#arrange () function

# Create a data frame with missing data

d <- data.frame( name = c("Abhi", "Bhavesh", "Chaman", "Dimri"),

age = c(7, 5, 9, 16),

ht = c(46, NA, NA, 69),

school = c("yes", "yes", "no", "no") )

# Arranging name according to the age

d.name<- arrange(d, age)

print(d.name)

**Output**

name age ht school

1 Bhavesh 5 NA yes

2 Abhi 7 46 yes

3 Chaman 9 NA no

4 Dimri 16 69 no

1. **Select ( ) and rename ( )**

For choosing variables and using their names as base for doing so.

**Script**

#Select () and rename ()

# Create a data frame with missing data

d <- data.frame(name=c("Abhi", "Bhavesh","Chaman", "Dimri"),

age=c(7, 5, 9, 16),

ht=c(46, NA, NA, 69),

school=c("yes", "yes", "no", "no"))

# startswith() function to print only ht data

select(d, starts\_with("ht"))

# -startswith() function to print

# everything except ht data

select(d, -starts\_with("ht"))

# Printing column 1 to 2

select(d, 1: 2)

# Printing data of column

# heading containing 'a'

select(d, contains("a"))

# Printing data of column

# heading which matches 'na'

select(d, matches("na"))

select(d, contains("a"))

**Output**

ht

1 46

2 NA

3 NA

4 69

name age school

1 Abhi 7 yes

2 Bhavesh 5 yes

3 Chaman 9 no

4 Dimri 16 no

name age

1 Abhi 7

2 Bhavesh 5

3 Chaman 9

4 Dimri 16

name age

1 Abhi 7

2 Bhavesh 5

3 Chaman 9

4 Dimri 16

name

1 Abhi

2 Bhavesh

3 Chaman

4 Dimri

name age

1 Abhi 7

2 Bhavesh 5

3 Chaman 9

4 Dimri 16

1. **mutate ( ) and transmute ( )**

Addition of new variables which are the functions of prevailing variables.

**Script**

#muatate () and transmute ()

# Create a data frame with missing data

d <- data.frame( name = c("Abhi", "Bhavesh",

"Chaman", "Dimri"),

age = c(7, 5, 9, 16),

ht = c(46, NA, NA, 69),

school = c("yes", "yes", "no", "no") )

# Calculating a variable x3 which is sum of height

# and age printing with ht and age

mutate(d, x3 = ht + age)

**Output**

name age ht school x3

1 Abhi 7 46 yes 53

2 Bhavesh 5 NA yes NA

3 Chaman 9 NA no NA

4 Dimri 16 69 no 85

x3

1 53

2 NA

3 NA

4 85

1. **summarize**

Condensing various values to one value is done by the summarize function.

**Script**

#Summarize()

# Create a data frame with missing data

d <- data.frame( name = c("Abhi", "Bhavesh", "Chaman", "Dimri"),

age = c(7, 5, 9, 16),

ht = c(46, NA, NA, 69),

school = c("yes", "yes", "no", "no") )

# Calculating mean of age

summarise(d, mean = mean(age))

# Calculating min of age

summarise(d, med = min(age))

# Calculating max of age

summarise(d, med = max(age))

# Calculating median of age

summarise(d, med = median(age))

**Output**

mean

1 9.25

med

1 5

med

1 16

med

1 8

1. **sample\_n( ) and sample\_frac( )**

For taking random specimens, sample\_n ( ) and sample\_frac ( ) are used.

**Script**

#sample\_n() and sample\_frac()

# Create a data frame with missing data

d <- data.frame( name = c("Abhi", "Bhavesh","Chaman", "Dimri"),

age = c(7, 5, 9, 16),

ht = c(46, NA, NA, 69),

school = c("yes", "yes", "no", "no") )

# Printing three rows

sample\_n(d, 3)

# Printing 50 % of the rows

sample\_frac(d, 0.50)

**Output**

name age ht school

1 Dimri 16 69 no

2 Abhi 7 46 yes

3 Chaman 9 NA no

name age ht school

1 Bhavesh 5 NA yes

2 Dimri 16 69 no

**RESULT**

Thus, the Data manipulation by ‘dplyr’ package has been done successfully in R.