|  |  |
| --- | --- |
| **EX.NO:2** | **WORKING WITH VECTORS AND MATRICES** |
| **DATE:** |

**OBJECTIVE:**

To create matrices and to manipulate the basic matrix operations

**SOFTWARE REQUIRED:**

R – Programming or R-studio Online

**Problem 1:** Create two vectors a and b of length 15 and find the following.

i) A+B vi) A-B

ii) 5A+6B vii) A/10

iii) 9A+5B viii) Square root of B

iv) 9A-15B ix) A/B

v) A+3 x) Square of A

**Algorithm:**

**Step1:** First Create 2 vectors of length 15.

**Step 2:** Manipulate the following results:

i) A+B vi) A-B

ii) (5\*A) + (6\*B) vii) A/10

iii) (9\*A) + (5\*B) viii) sqrt(B)

iv) (9\*A) - (15\*B) ix) A/B

v) A+3 x) A\*A

**Step 3:** Stop the process

**Input:**

> a=seq(2,30,by=2)

> b=seq(3,45,by=3)

>a+b

> (5\*a)+(6\*b)

> (9\*a)+(5\*b)

> (9\*a)-(15\*b)

>a-b

>a/10

>sqrt(b)

>a/b

>a+3

>a^2

>nchar(a)

>length(a)

>length(b)

**Output:**

> a=seq(2,30,by=2)

>a

[1] 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30

> b=seq(3,45,by=3)

>b

[1] 3 6 9 12 15 18 21 24 27 30 33 36 39 42 45

>a+b

[1] 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75

> (5\*a)+(6\*b)

[1] 28 56 84 112 140 168 196 224 252 280 308 336 364 392 420

> (9\*a)+(5\*b)

[1] 33 66 99 132 165 198 231 264 297 330 363 396 429 462 495

> (9\*a)-(15\*b)

[1] -27 -54 -81 -108 -135 -162 -189 -216 -243 -270 -297 -324 -351 -378 -405

>a-b

[1] -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14 -15

>a/10

[1] 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0

>sqrt(b)

[1] 1.732051 2.449490 3.000000 3.464102 3.872983 4.242641 4.582576 4.898979 5.196152 5.477226 5.744563 6.000000 6.244998 6.480741 6.708204

>a/b

[1] 0.6666667 0.6666667 0.6666667 0.6666667 0.6666667 0.6666667 0.6666667 0.6666667 0.6666667 0.6666667 0.6666667 0.6666667 0.6666667 0.6666667 0.6666667

>a+3

[1] 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33

>a^2

[1] 4 16 36 64 100 144 196 256 324 400 484 576 676 784 900

> #length of a

>nchar(a)

[1] 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2

>length(a)

[1] 15

> #length of b

>length(b)

[1] 15

**Problem 2:**Create vector of 10 strings and find the length of strings.

**Algorithm:**

**Step 1:** First create a vector to store 10 strings, individual strings are enclosed within double quotes.

**Step 2:** Use **nchar(vector\_name)** to display the length of the respective strings.

**Step 3:** Stop the process.

**Input:**

>A=c("KABILAN","MOKESH","YUSUF","HACKING","VIJAY","VENKY","STAT","MODEL","RLAN","MATRIX")

>nchar(A)

**Output:**

>A=c("KABILAN","MOKESH","YUSUF","HACKING","VIJAY","VENKY","STAT","MODEL","RLAN","MATRIX")

>nchar(A)

[1] 7 6 5 7 5 5 4 5 4 6

**Problem3:** Create a 3 x 3 matrix and verify the commutative and associative laws for addition and multiplication.

**Algorithm:**

**Step1:** First create 3 3 x 3 matrices either by using range or user-defined values(vectors).

**Step 2:** Manipulate the following results for **Addition:**

**Commutative:**

A+B=B+A

` **Associative:**

A+(B+C) =(A+B) +C

**Step 3:** Manipulate the following results for **Multiplication:**

**Commutative:**

A\*B = B\*A (Element wise multiplication)

A%\*%B = B%\*%A (Matrix Multiplication)

**Associative:**

(A\*B) \*C = A \*(B\*C) (Element wise multiplication)

(A%\*%B) %\*%C = A%\*%(B%\*%C) (Matrix Multiplication)

**Step 4:** Stop the process

**Input:**

> A=matrix(seq(4,36,by=4),nrow=3,ncol=3)

> B=matrix(seq(6,54,by=6),nrow=3,ncol=3)

> C=matrix(seq(2,18,by=2),nrow=3,ncol=3)

> A+B

> B+A

> A\*B

> B\*A

> (A+B)+C

> A+(B+C)

> A\*(B\*C)

> (A\*B)\*C

>A%\*%(B%\*%C)

> (A%\*%B)%\*%C

**Output:**

> # To verify commutative and associative for 3X3 matrix

> A=matrix(seq(4,36,by=4),nrow=3,ncol=3)

> A

[,1] [,2] [,3]

[1,] 4 16 28

[2,] 8 20 32

[3,] 12 24 36

> B=matrix(seq(6,54,by=6),nrow=3,ncol=3)

> B

[,1] [,2] [,3]

[1,] 6 24 42

[2,] 12 30 48

[3,] 18 36 54

> C=matrix(seq(2,100,by=2),nrow=3,ncol=3)

> C

[,1] [,2] [,3]

[1,] 2 8 14

[2,] 4 10 16

[3,] 6 12 18

**> A+B**

**[,1] [,2] [,3]**

**[1,] 10 40 70**

**[2,] 20 50 80**

Commutative for addition

**[3,] 30 60 90**

**> B+A**

**[,1] [,2] [,3]**

**[1,] 10 40 70**

**[2,] 20 50 80**

**[3,] 30 60 90**

**> A\*B**

**[,1] [,2] [,3]**

**[1,] 24 384 1176**

**[2,] 96 600 1536**

**[3,] 216 864 1944**

Commutative for multiplication

**> B\*A**

**[,1] [,2] [,3]**

**[1,] 24 384 1176**

**[2,] 96 600 1536**

**[3,] 216 864 1944**

> #matrix multiplication

**> A%\*%B**

**[,1] [,2] [,3]**

**[1,] 720 1584 2448**

**[2,] 864 1944 3024**

**[3,] 1008 2304 3600**

Commutative for matrix multiplication

**> B%\*%A**

**[,1] [,2] [,3]**

**[1,] 720 1584 2448**

**[2,] 864 1944 3024**

**[3,] 1008 2304 360**

**> A+(B+C)**

**[,1] [,2] [,3]**

**[1,] 12 48 84**

**[2,] 24 60 96**

**[3,] 36 72 108**

Associative for addition

**> (A+B)+C**

**[,1] [,2] [,3]**

**[1,] 12 48 84**

**[2,] 24 60 96**

**[3,] 36 72 108**

**> A\*(B\*C)**

**[,1] [,2] [,3]**

**[1,] 48 3072 16464**

**[2,] 384 6000 24576**

**[3,] 1296 10368 34992**

Associative for element multiplication

**> (A\*B)\*C**

**[,1] [,2] [,3]**

**[1,] 48 3072 16464**

**[2,] 384 6000 24576**

**[3,] 1296 10368 34992**

**>A%\*%(B%\*%C)**

**[,1] [,2] [,3]**

**[1,] 22464 50976 79488**

**[2,] 27648 62640 97632**

**[3,] 32832 74304 115776**

Associative for matrix multiplication

**> (A%\*%B)%\*%C**

**[,1] [,2] [,3]**

**[1,] 22464 50976 79488**

**[2,] 27648 62640 97632**

**[3,] 32832 74304 115776**

**>**

**Problem 4: C**reate a matrix of order 5 x 5 and manipulate the following:

* + - Inverse of the matrix
    - Determinant of the matrix
    - Transpose of the matrix

**Algorithm:**

**Step 1:** First create a 5 x 5 matrix consisting of user defined values with the help of vectors (either row bind or column bind).

**Step 2:** Use either the command **rbind (**for row bind) or **cbind (**for column bind) to bind the vectors as a matrix.

**Step 3:** Use **solve(matrix\_variable)** to find the **Inverse of the matrix.**

**Step 4:** Use **det(matrix\_variable)** to find the **determinant of the matrix.**

**Step 5:** Use **t(matrix\_variable)** to find the **transpose of the matrix.**

**Step 6:** Stop the process

**Input:**

> row1=c(5,8,6,8,7)

> row2=c(4,-6,3,7,6)

> row3=c(1,8,10,14,6)

> row4=c(7,-7,4,3,2)

> row5=c(9,18,6,7,5)

>mat=rbind(row1,row2,row3,row4,row5)

>solve(mat)

>det(mat)

>t(mat)

**Output:**

>mat

[,1] [,2] [,3] [,4] [,5]

row1 5 8 6 8 7

row2 4 -6 3 7 6

row3 1 8 10 14 6

row4 7 -7 4 3 2

row5 9 18 6 7 5

>**solve(mat)**

**row1 row2 row3 row4 row5**

**[1,] -0.148810528 0.08334739 -0.028626062 0.04566672 0.12440245**

**[2,] 0.001181036 -0.01653450 -0.002418312 -0.03739947 0.03604972**

**[3,] 0.382824363 -0.35954108 0.025645352 0.21056184 -0.21950396**

**[4,] -0.448962376 0.28547326 0.109780102 -0.11619144 0.20071987**

**[5,] 0.432765311 -0.05871436 -0.124233733 -0.03756819 -0.17130645**

>**det(mat)**

**[1] -17781**

>**t(mat)**

**row1 row2 row3 row4 row5**

**[1,] 5 4 1 7 9**

**[2,] 8 -6 8 -7 18**

**[3,] 6 3 10 4 6**

**[4,] 8 7 14 3 7**

**[5,] 7 6 6 2 5**

**Practice Problem:**

**1.** Create 2 vectors of length 10 and manipulate the following operations

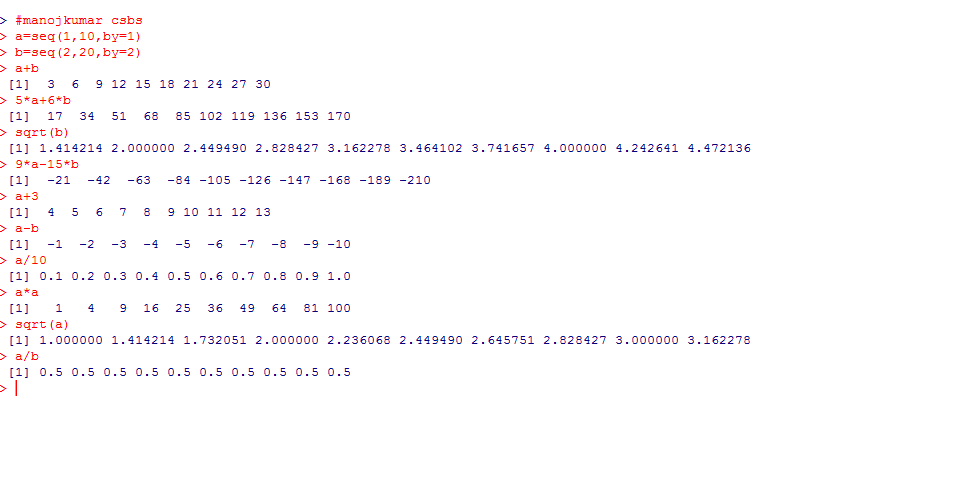
i) A+B vi) A-B

ii) 5A+6B vii) A/10

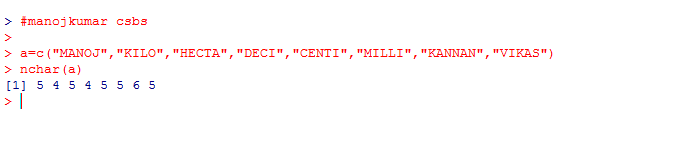
iii) Square root of B viii) Square root of A

iv) 9A-15B ix) A/B

v) A+3 x) Square of A

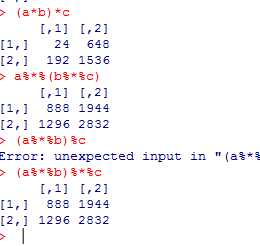


2. Create a vector of 8 strings and calculate the length of the individual strings



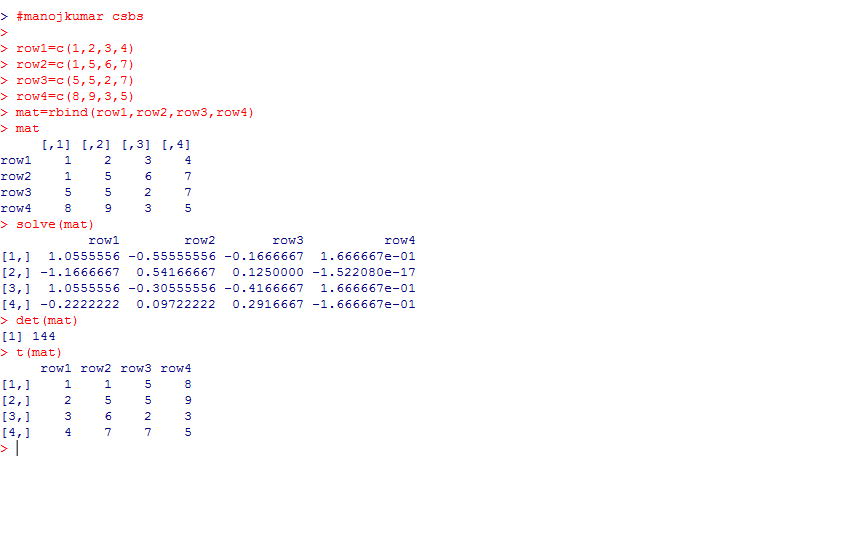
3. Create a2 x 2 matrix and verify the commutative and associative laws for addition and multiplication.





4. Create a matrix of order 4 x 4 and manipulate the following:

* + - Inverse of the matrix
    - Determinant of the matrix
    - Transpose of the matrix



**Conclusion**

Thus, the R program to create and manipulate various vectors and matrices is successful and the output is displayed accordingly.