Task-2_Matrix-Operations.R

vishnu

2023-11-06

```
#Matrix Operations

#1.Create two matrices, matrix_A and matrix_B
matrix_A <- matrix(c(4,10,5,8,16,9,23,45,16),nrow = 3,ncol = 3,byrow = TRUE,dimnames = list(c
("A","B","C"),c("X","Y","Z")))
matrix_A
```

```
##    X  Y  Z
## A  4 10  5
## B  8 16  9
## C 23 45 16
```

```
matrix_B <- matrix(c(4,5,6,1,1,3,7,8,9),nrow = 3,ncol = 3,byrow = TRUE,dimnames = list(c
("A","B","C"),c("X","Y","Z")))
matrix_B
```

```
##   X Y Z
## A 4 5 6
## B 1 1 3
## C 7 8 9
```

```
#2.Calculate the sum of matrix_A and matrix_B and store the result in a new matrix named matr
ix_sum.
matrix_sum = matrix_A+matrix_B
matrix_sum
```

```
##    X  Y  Z
## A  8 15 11
## B  9 17 12
## C 30 53 25
```

```
#3.Calculate the difference between matrix_A and matrix_B and store the result in a new matri
x named matrix_diff.
#difference of matrix's
matrix_diff = matrix_A-matrix_B
matrix_diff
```

```
##    X  Y  Z
## A  0  5 -1
## B  7 15  6
## C 16 37  7
```

```
#4.Multiply matrix_A by a scalar value of 2 and store the result in a new matrix named matrix
_mul
matric_mult = matrix_A*2
matric_mult
```

```
##    X  Y  Z
## A  8 20 10
## B 16 32 18
## C 46 90 32
```

```
#5.Calculate the product of matrix_A and matrix_B and store the result in a new matrix named
matrix_product.
matrix_product = matrix_A*matrix_B
matrix_product
```

```
##     X   Y   Z
## A  16  50  30
## B   8  16  27
## C 161 360 144
```

```
#6.Find the transpose of matrix_A and store the result in a new matrix named matrix_A_transpo
se.
matrix_A_transpose=t(matrix_A)
matrix_A_transpose
```

```
##    A  B  C
## X  4  8 23
## Y 10 16 45
## Z  5  9 16
```

```
#7.Calculate the determinant of matrix_B and store it in a variable named determinant_B.
determinant_B=det(matrix_B)
determinant_B
```

```
## [1] 6
```

```
#8.Invert matrix_B to obtain the inverse matrix and store it in a new matrix named matrix_B_i
nverse.
matrix_B_inverse= solve(matrix_B)
matrix_B_inverse
```

```
##            A    B          C
## X -2.5000000  0.5  1.5000000
## Y  2.0000000 -1.0 -1.0000000
## Z  0.1666667  0.5 -0.1666667
```

```
#9.Check if matrix_B is orthogonal (i.e., its transpose is equal to its inverse).
matrix_B_transpose=t(matrix_B)
matrix_B_transpose==matrix_B_inverse
```

```
##       A     B     C
## X FALSE FALSE FALSE
## Y FALSE FALSE FALSE
## Z FALSE FALSE FALSE
```

```
#10.Calculate the element-wise square root of matrix_A and store the result in a new matrix n
amed matrix_A_sqrt
matrix_A_sqrt =  sqrt(matrix_A)
matrix_A_sqrt
```

```
##          X        Y        Z
## A 2.000000 3.162278 2.236068
## B 2.828427 4.000000 3.000000
## C 4.795832 6.708204 4.000000
```

```
#11.Calculate the mean of all the elements in matrix_B.
print(mean(matrix_B))
```

```
## [1] 4.888889
```

```
#12.Calculate the sum of each column in matrix_A.
sum_matrix_A = c(sum(matrix_A[,1]),sum(matrix_A[,2]),sum(matrix_A[,3]))
sum_matrix_A
```

```
## [1] 35 71 30
```

```
#13.Calculate the row means of matrix_B.
row_means_matrix_B <- c(mean(matrix_B[1,]),mean(matrix_B[2,]),mean(matrix_B[3,]))
row_means_matrix_B
```

```
## [1] 5.000000 1.666667 8.000000
```

```
#14.Extract the second row of matrix_A and store it in a vector named second_row_A.
second_row_A <- c(matrix_A[2,])
second_row_A
```

```
##  X  Y  Z
##  8 16  9
```

```
#15.Extract the third column of matrix_B and store it in a vector named third_column_B.
third_column_B <- matrix_B[,3]
third_column_B
```

```
## A B C
## 6 3 9
```