Task2–Vector-Operations.R

aaron

2023-11-06

```
#1.Create two matrices, matrix_A and matrix_B
matrix_A=matrix(c(4,6,1,2,5,2,8,4,9),nrow = 3,ncol = 3,byrow = TRUE);matrix_A
```

```
##      [,1] [,2] [,3]
## [1,]    4    6    1
## [2,]    2    5    2
## [3,]    8    4    9
```

```
matrix_B=matrix(c(6,2,1,2,3,8,5,0,8),nrow = 3,ncol = 3,byrow = TRUE);matrix_B
```

```
##      [,1] [,2] [,3]
## [1,]    6    2    1
## [2,]    2    3    8
## [3,]    5    0    8
```

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

```
##      [,1] [,2] [,3]
## [1,]   10    8    2
## [2,]    4    8   10
## [3,]   13    4   17
```

```
#3.Calculate the difference between matrix_A and matrix_B and store the result in
a new matrix named matrix_diff.
matrix_diff =  matrix_A-matrix_B;matrix_diff
```

```
##      [,1] [,2] [,3]
## [1,]   -2    4    0
## [2,]    0    2   -6
## [3,]    3    4    1
```

```
#4.Multiply matrix_A by a scalar value of 2 and store the result in a new matrix n
amed matrix_mult.
matrix_mult = matrix_A * 2;matrix_mult
```

```
##      [,1] [,2] [,3]
## [1,]    8   12    2
## [2,]    4   10    4
## [3,]   16    8   18
```

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

```
##      [,1] [,2] [,3]
## [1,]   41   26   60
## [2,]   32   19   58
## [3,]  101   28  112
```

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

```
##      [,1] [,2] [,3]
## [1,]    4    2    8
## [2,]    6    5    4
## [3,]    1    2    9
```

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

```
## [1] 177
```

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

```
##             [,1]        [,2]        [,3]
## [1,]  0.13559322 -0.09039548  0.07344633
## [2,]  0.13559322  0.24293785 -0.25988701
## [3,] -0.08474576  0.05649718  0.07909605
```

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

```
## [1] FALSE
```

```
if (is_orthogonal) {
  print("Matrix_B is orthogonal.")
} else {
  print("Matrix_B is not orthogonal.")
}
```

```
## [1] "Matrix_B is not orthogonal."
```

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

```
##          [,1]     [,2]     [,3]
## [1,] 2.000000 2.449490 1.000000
## [2,] 1.414214 2.236068 1.414214
## [3,] 2.828427 2.000000 3.000000
```

```
#11.Calculate the mean of all the elements in matrix_B.
mean_matrix_B = mean(matrix_B);mean_matrix_B
```

```
## [1] 3.888889
```

```
#12.Calculate the sum of each column in matrix_A.
sum_columns_A= colSums(matrix_A);sum_columns_A
```

```
## [1] 14 15 12
```

```
#13.Calculate the row means of matrix_B.
mean_row_matrix_B = rowMeans(matrix_B);mean_row_matrix_B
```

```
## [1] 3.000000 4.333333 4.333333
```

```
#14.Extract the second row of matrix_A and store it in a vector named second_row_
A.
second_row_A = matrix_A[2, ];second_row_A
```

```
## [1] 2 5 2
```

```
#15.Extract the third column of matrix_B and store it in a vector named third_colu
mn_B.
third_column_B = matrix_B[, 3];third_column_B
```

```
## [1] 1 8 8
```