Matrix Operation

Matrix Multiplication

matrix multiplication is a binary operation that produces a matrix from two matrices. For matrix multiplication, the number of columns in the first matrix must be equal to the number of rows in the second matrix.

```
# Matrix multiplication
mat_mul <- function(mat1, mat2) {
  if (dim(mat1)[2] == dim(mat2)[1]) {
    c = matrix(0, nrow = dim(mat1)[1], ncol = dim(mat2)[2])
    for (i in 1:dim(mat1)[1]) {
      for (j in 1:dim(mat2)[2]) {
        for (k in 1:dim(mat1)[2]) {
          c[i,j] = c[i,j] + mat1[i,k]*mat2[k,j]
        }

      }

    }
  }else{
    return('1st Matrix column number and 2nd matrix row number not equal')
  }
  return(c)
}

A = matrix(c(1,2,3,4,5,6), nrow = 2)
B = matrix(c(seq(10,18,1)), nrow = 3)
mat_mul(A,B)
```

```
##      [,1] [,2] [,3]
## [1,]  103  130  157
## [2,]  136  172  208
```

```
mat_mul(B,A)
```

```
## [1] "1st Matrix column number and 2nd matrix row number not equal"
```

transpose of a matrix

The transpose of a matrix is found by interchanging its rows into columns or columns into rows.

```
transpose <- function(mat1) {
  mat = matrix(0, nrow = dim(mat1)[2], ncol = dim(mat1)[1])
  for (i in 1:dim(mat1)[2]) {
    for (j in 1:dim(mat1)[1]) {
      mat[i,j] = mat1[j,i]

    }

  }
  return (mat)
}

transpose(A)
```

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

Matrix trace

The trace of a matrix is the sum of the diagonal elements of the matrix the trace of a matrix only exists in the matrix is a square matrix. In this case, is not square. Therefore, the trace does not exist.

```
trace <- function(mat1) {
  if (dim(mat1)[1] == dim(mat1)[2] ){
    count = 0
    for (i in 1:dim(mat1)[1]){
      for (j in 1:dim(mat1)[2]){
        if (i == j){
          count = count + mat1[i,j]
        }
      }
    }
  }else{
    return ('Non Square matrix Trace does not exist')
  }
  return (count)
}

trace(B)
```

```
## [1] 42
```

```
trace(A)
```

```
## [1] "Non Square matrix Trace does not exist"
```

Off diagonal element sum

```
off_dsum <- function(mat1) {
  count = 0
  for (i in 1:dim(mat1)[1]){
    for (j in 1:dim(mat1)[2]){
      if(i != j){
        count = count + mat1[i,j]
      }
    }
  }
  return (count)
}
off_dsum(B)
```

```
## [1] 84
```

matrix elements sum

```
mat_sum <- function(mat1) {
  count = 0
  for(i in 1:dim(mat1)[1]){
    for(j in 1:dim(mat1)[2]){
      count = count + mat1[i,j]
    }
  }
  return(count)
}
mat_sum(B)
```

```
## [1] 126
```

Symmetric matrix

```
mat_sym <- function(mat1) {
  row = dim(mat1)[1]
  col = dim(mat1)[2]
  if(row == col){
    for(i in 1:row){
      for(j in 1:col){
       if(i != j){
         if(mat1[i,j] == mat1[j,i]){
           return ('Yes this is a Symmetric Matrix')
         }else{
           return('No this is not a symmetric Matrix')
         }
       }
      }
    }
  }else{
    return('This is not a square matrix')
  }
}
mat_sym(A)
```

```
## [1] "This is not a square matrix"
```

```
mat_sym(B)
```

```
## [1] "No this is not a symmetric Matrix"
```

```
C = matrix(c(1,2,3,2,3,4,3,4,5), nrow = 3, byrow = T)
C
```

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

```
mat_sym(C)
```

```
## [1] "Yes this is a Symmetric Matrix"
```