

Illustration of Code

This code basically calculates the inverse of matrix and what does more is interesting, it uses the cached data to increasing the processing speed i.e it checks if the inverse of matrix is already present in the cached memory. If so then it uses it otherwise it calculates it.

Functions

This function makes the inverse of matrix stored in cached memory of R and returns the cached data if present

```
makeCacheMatrix <- function(matrix = matrix()) {  
  inverse_of_matrix <- NULL  
  setmatrix <- function(y){  
    matrix<<- y  
    # since we are changing the matrix , hence we initialize the inverse of matrix as null  
    inverse_of_matrix<<-NULL  
  }  
  
  getmatrix <- function() matrix  
  set_inverse_of_matrix <- function(inverse_of_matrix) inverse_of_matrix<<- inverse_of_matrix  
  get_inverse_of_matrix <- function() inverse_of_matrix  
  list(setmatrix=setmatrix,getmatrix=getmatrix,set_inverse_of_matrix=set_inverse_of_matrix,  
       get_inverse_of_matrix=get_inverse_of_matrix)  
}
```

What this function do is basically check if the inverse is already present in cached memory or not , and if it is present it returns it, otherwise it calculates the inverse of matrix.

```
cacheSolve <- function(x, ...) {  
  inverse <- x$get_inverse_of_matrix()  
  if(!is.null(inverse)){  
    message("Using the cached data and returning it")  
    return(inverse)  
  }  
  else{  
    data <- x$getmatrix()  
    inverse<- solve(data,...)  
    x$set_inverse_of_matrix(inverse)  
    inverse  
  }  
}
```

```

fun <- makeCacheMatrix()
a <- matrix(c(2,42,6,22),nrow=2,ncol=2)
fun$setmatrix(a)
# Checking if matrix is stored
fun$getmatrix()

```

```

##      [,1] [,2]
## [1,]    2    6
## [2,]   42   22

```

```
cacheSolve(fun)
```

```

##      [,1]      [,2]
## [1,] -0.1057692  0.028846154
## [2,]  0.2019231 -0.009615385

```

```
cacheSolve(fun)
```

```
## Using the cached data and returning it
```

```

##      [,1]      [,2]
## [1,] -0.1057692  0.028846154
## [2,]  0.2019231 -0.009615385

```

This time the cached data is used