R P...g...g ...

Thymios C

22 October 2016

Assignment: Caching the

Matrix inversion is usually a costly computation and there repeatedly (there are also alternatives to matrix inversion the inverse of a matrix.

Write the following functions:

makeCacheMatrix: This function creates a special "matrix the special "matrix" returned by makeCacheMatrix above cachesolve should retrieve the inverse from the cache. C example, if X is a square invertible matrix, then solve(X) re

For this assignment, assume that the matrix supplied

The following functions are used to create a special objec creates a special "matrix", which is really a list containing

1. set the value of the matrix
2. get the value of the matrix
3. set the value of the inverse
4. get the value of the inverse

```
makeCacheMatrix <- function(x = matrix()) {
    i <- NULL
  set <- function(y) {
          x <<- y
          i <<- NULL
  }
  get <- function() x
  setinverse <- function(inverse) i <<- inve
  getinverse <- function() i
  list(set = set,
       get = get,
       setinverse = setinverse,
       getinverse = getinverse)
}
```

This function computes the inverse of the special "matrix the matrix has not changed), then cacheSolve should ret

```
cacheSolve <- function(x, ...) {
  i <- x$getinverse()
  if (!is.null(i)) {
```