cachematrix.R

NMac

Sat May 23 21:02:09 2015

## The following functions create and manipulate   
## a cached matrix and its inverse.  
  
  
## makeCacheMatrix creates a matrix by passing a   
## variety of data and optional parameters.  
makeCacheMatrix <- function(x=matrix()) {  
 i <- NULL ## initializes i  
 set <- function(dat, ...) { ## input data/params   
 x <<- matrix(data=dat, ...) ## creates matrix  
 i <<- NULL ## stores inv &/or signals new values  
 }  
 get <- function() x ## retrieves user-created matrix  
 setinv <- function(ivx) i <<- ivx ## insert calc'd inv  
 getinv <- function() i ## retrieve calculated inv  
 list(set=set, get=get, setinv=setinv, ## store func  
 getinv=getinv) ## "  
}  
  
## cacheSolve creates & caches an inverse matrix from   
## the user-defined matrix. It also retrieves a cached   
## inverse matrix where one already exists.  
cacheSolve <- function(x, ...) {  
 im <- x$getinv() ## retrieve value from cached inv matrix  
 if(!is.null(im)) { ## eval's for existing inv matrix  
 message("getting cached data") ## inv already cached  
 return(invisible(im)) ## return existing inv matrix  
 ## made invisible since redundant with getinv()  
 }  
 data <- x$get() ## (if no existing inv) retrieve orig matrix  
 im <- solve(data) ## create inv matrix from user-def'd matrix  
 x$setinv(im) ## pass inv matrix back to makeCacheMatrix  
 invisible(im) ## returns inv matrix as good practice  
 ## again, invisible, since redundant w/ getinv()  
}