## Assignment: Caching the Inverse of a Matrix

## This page includes a pair of functions that cache the inverse of a matrix.

##makeCacheMatrix: This function creates a special "matrix" object that can cache its inverse.

makeCacheMatrix <- function(x = matrix()) {

i <- NULL

set <- function(y) {

x <<- y

i <<- NULL

}

get <- function() x

setinverse <- function(inverse) i <<- inverse

getinverse <- function() i

list(set = set,

get = get,

setinverse = setinverse,

getinverse = getinverse)

}

#cacheSolve: This function computes the inverse of the special "matrix" returned by makeCacheMatrix above.

#If the inverse has already been calculated (and the matrix has not changed),

#then the cachesolve should retrieve the inverse from the cache.

cacheSolve <- function(x, ...) {

i <- x$getinverse()

if (!is.null(i)) {

message("getting cached data")

return(i)

}

data <- x$get()

i <- solve(data, ...)

x$setinverse(i)

i

}