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

## This function creates a special “matrix” object that can cache its inverse

+ inv <- NULL

+ set <- function(y){

+ x <<- y

+ inv <<- NULL

+ }

+ get <- function() x

+ setInverse <- function(solveMatrix) inv <<- solveMatrix ## assigns value of inv in parent environment

+ getInverse <- function() inv ## gets the value of inv where called

+ list(set = set, get = get, setInverse = setInverse, getInverse = getInverse)

+ }

>

>

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

+ ## Return a matrix that is the inverse of 'x'

+ ## Inverse of matrix

+ inv <- x$getInverse()

+ if(!is.null(inv)){

+ message("getting cached data")

+ return(inv)

+ }

+ data <- x$get()

+ ##using solve() function to get inverse of matrix

+ inv <- solve(data)

+ x$setInverse(inv)

+ inv

+ }