The first function, makeVector creates a special “vector”, which is really a list containing a function to

set the value of the vector

get the value of the vector

set the value of the mean

get the value of the mean

makeVector <- function(x = numeric()) {

m <- NULL

set <- function(y) {

x <<- y

m <<- NULL

}

get <- function() x

setmean <- function(mean) m <<- mean

getmean <- function() m

list(set = set, get = get,

setmean = setmean,

getmean = getmean)

}

The following function calculates the mean of the special “vector” created with the above function. However, it first checks to see if the mean has already been calculated. If so, it gets the mean from the cache and skips the computation. Otherwise, it calculates the mean of the data and sets the value of the mean in the cache via the setmean function

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

m <- x$getmean()

if(!is.null(m)) {

message("getting cached data")

return(m)

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

j <- NULL

set <- function(y){

x <<- y

j <<- NULL

}

get <- function()x

setInverse <- function(inverse) j <<- inverse

getInverse <- function() j

list(set = set, get = get,

setInverse = setInverse,

getInverse = getInverse)

}

##Please include your own comment to explain your code (Required in Rubric)

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

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

j <- x$getInverse()

if(!is.null(j)){

message("getting cached data")

return(j)

}

mat <- x$get()

j <- solve(mat,...)

x$setInverse(j)

  j

}data <- x$get()

m <- mean(data, ...)

x$setmean(m)

        m

}