

Assignment two

DanielH

July 6, 2017

Function one

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

```
makeCacheMatrix <- function(x = matrix()) {  
  inv <- NULL  
  set <- function(y) {  
    x <- y  
    inv <- NULL  
  }  
  get <- function() x  
  setInverse <- function(inverse) inv <- inverse  
  getInverse <- function() inv # Value of the matrix inverse  
  list(set = set, # list of the functions  
        get = get,  
        setInverse = setInverse,  
        getInverse = getInverse)  
}
```

Function two

This function computes the inverse of the special “matrix” created by the makeCacheMatrix function above. If the inverse has already been calculated (and the matrix has not changed), the following function retrieves the inverse from the cache.

```
cacheSolve <- function(x, ...) {  
  ## Return a matrix that is the inverse of 'x'  
  inv <- x$getInverse()  
  if (!is.null(inv)) {  
    message("getting cached data")  
    return(inv)  
  }  
  mat <- x$get()  
  inv <- solve(mat, ...)  
  x$setInverse(inv)  
  inv  
}
```