Peer Review - Hyunsang Son

One of the big take aways from solving a system $A\beta = B$ was the difference between: `solve(A) % * % B` and `solve(A,B)` . The inversion method uses the first expressions, where you solve for the inverse of A directly, and then multiply it by B. The second method should always be faster then the inversion method. Any optimal performance you saw from the inversion method was due to you needlessly inverting matrices in you other methods. By default R uses QR decomposition, so you can think of the second method as an optimized QR decomposition. For the implementation of the Cholesky decomposition, you shouldn't need to use `solve()` at all. Instead, you can take the Cholesky decomposition and use `backsolve()` and `forwardsolve()` . My code for the Cholesky decomposition is as follows:

```
 Cholesky.method <- function(a,b){temp <- chol(a)
Z <- backsolve(temp,b,transpose=TRUE)
backsolve(temp,Z)}
```

This should be faster than the inversion method, and in my code was faster than the generic `solve(A,B)` .

For sparse matrices the *Matrix* package allows you to define matrices as sparse. Once the matrices are defined as sparse, you can use the same functions you used before, such as `solve()` , but they will reprogrammed to efficiently deal with sparse matrices.

Other than those notes, I would say ask your fellow students for help, and maybe try to collaborate, or get together to go over homework assignments.