In NAMESPACE of the package, you need to add

```
useDynLib(normalregMixMay09)
```

which allows to import all compiled functions of the package itself; note that normalregMixMay09 is the name of a temporary package I have created for testing in this directory. You might also want to use the following command in command prompt to make sure that all codes are running properly:

```
R CMD check normalregMixMay09_0.1.0.tar.gz
```

At this current stage, only one error should appear due to improper documentation. If useDynLib is not specified in the namespace of the package, apparently two additional tests appear when it attempts to run a few tests on check; to be specific, when it tries to call the codes written in C. I believe this package I have created should also work in Mac OS too, as useDynLib is not an OS-specific operation. Even if that's the case, I have uploaded this package to check for compatibility.

Troubleshooting

If the line `R CMD check normalregMixMay09_0.1.0.tar.gz` does not work, try

1. Install RTools.
2. Add the followings on path:

   ```
   C:\Program␣Files\R\R-3.2.5\bin\x64
   C:\RBuildTools\3.3\bin
   C:\RBuildTools\3.3\gcc-4.6.3\bin
   ```

   Note that the directory might be different for RTools; in my computer, it was under the directory called `RBuildTools`, not `RBuild`, which seems to be default for a number of people.

Additional Note on Loading C/C++ Codes

It turns out that library files created in compilation are not compatible between Mac OS and Windows. If you want to compile and create a new package, make sure that you **remove** existing

`.o` and `.so` files in the src folder. The same principle applies when you commit to a git; remove them in prior to submission.