It is well known that debugging distributed programs is more difficult than doing the same on local, sequential ones. This is some suggestions to make debugging rmr programs more accessible.

- Start with the local backend. It should run your code unmodified (main exceptions are I/O formats that require java classes, those are not available with the local backend) and is a normal R sequential program. You can debug it the normal way, for instance using `debug()`
- Once your program runs in local mode, switch to hadoop mode. Depending on your hadoop configuration, hadoop can run in three modes, standalone, pseudo-distributed and distributed. You should learn how to quickly switch between the three and continue your debugging in standalone mode. In standalone, R errors are reported in console, that is in your regular R environment.
- Once your program run with the Hadoop backend with hadoop in standalone, you are ready to switch to pseudo-distributed or distributed modes. In these two modes, to find R errors you have to dig out the logs, specifically those called "userlogs". A good guide to hadoop log files has been put together by Cloudera. It is a little dated but the best I know of.
- In parallel, you should grow your test data set sizes. Small data sets let you go through debugging iterations faster and the local backend loads everything in main memory, so that's another limit to data set sizes. The largest data sets can trigger different bugs and can be processed only in distributed mode in a reasonable amount of time. `to.dfs()` is your friend in generating data.
- With the hadoop backend, you can not use `debug()` on the mapper and reducer, it has no effect. But you can still gather information on what is going on there with print statements. I normally use something like `cat(var1, var2, ..., file = stderr())`. The output of those statements will end up in console in standalone mode and in the userlogs in pseudo-distributed and distributed modes, just as we've seen for errors. It is very important that you do not use stdin and stdout in the mapper or reducer as they are reserved for communication between R and hadoop. Be sure to remove or inactivate these statements once running in production.
- If you suspect the map function is not doing what you want, there is a simple way to inspect the output of the map phase, just remove the reducer (leave it to default) and inspect the output of the job without the reducer. Maybe obvious to most of you but just in case.