Assignment 10

1. The objective of this problem is to give you some experience running simple parallel computations in R. You will use the `parallel` package for this. The manual for the `package` is available at

   `http://stat.ethz.ch/R-manual/R-devel/library/parallel/html/00Index.html`.

   The package vignette may also be useful:

   `http://stat.ethz.ch/R-manual/R-devel/library/parallel/doc/parallel.pdf`.

   A simple example along the lines of this problem is given in

   `http://homepage.stat.uiowa.edu/~luke/classes/STAT7400/examples/parallel/gamma.R`.

   Some examples illustrating how to pass function and variable definitions from the master to the workers and loading packages on the workers are given in

   `http://homepage.stat.uiowa.edu/~luke/classes/STAT7400/examples/parallel/export.R`.

   (a) Returning to the setting or Problem 2 from Assignment 7, write a function that takes the number of simulation replications `R`, the sample size `n` for the data sets ($n = 50$ is what you used previously), and the error standard deviation `s` as arguments and returns a matrix with one row for each x_i and one column for the estimated bias and estimates standard error at each x_i for the fit produced by the `gam` estimator in the `mgcv` package.

   (b) Write a function that takes a computational cluster as its first argument, followed by the arguments of the previous function, splits the simulation across the cluster, and returns a result in the same form as the previous function.

   (c) Use your functions for $n = 50$, $\sigma = 0.2$, and $R = 10,000$ on a cluster of at least 2 workers and confirm that you are getting a a significant reduction in elapsed time using the parallel approach. Also run your simulation twice with the same seed and confirm that you are getting identical results.

You should submit your assignment electronically using Icon. Submit your work as a single compressed tar file. If your work is in a directory `mywork` then you can create a compressed tar file with the command

```
tar czf mywork.tar.gz mywork
```

Solutions and Comments

1. Some notes:

   - You should shut down your cluster using `stopCluster`.
   - Make sure your function signature matches the specification.
   - For any parallel simulation it is a good idea to use a parallel RNG.
   - When parallelising code using a distributed memory approach:
     - It is a goo idea to put as much into your parallel function as possible.
     - This allows you to debug and test as much of the code a possible in a sequential setting.
     - It is also important to limit the amount of data that has to be transferred. So perform data reductions on the worker as much as possible.
   - The objects returned by `makeCluster` represent a set of separate R processes.
   - By default, these processes communicate with the master over *sockets*.
   - Other communication mechanisms are possible (e.g. MPI).
   - This framework supports parallel computing using one or more scatter-compute-gather operations.
   - Much more sophisticated parallel algorithms are possible but are much harder to learn to program. The `Rmpi` package might be useful for implementing some of these.
   - Nevertheless, many problems can be usefully parallelized with just scatter-compute-gather steps.
   - If you need to use a sequence of such steps you may need to minimize the amount of data communicated between workers and master between the steps for the parallelization to be effective.
   - The `snow` package provides some timing and visualization tools for understanding how much time is spent in communication.
   - A very major problem in general parallel computing is the possibility of deadlock. This will not happen in this simple framework.
   - It is possible to start worker jobs running that either do not terminate or would take too long to wait for. At the moment these have to be terminated by external means.