Lab 09 STATS 769 Due 4:00pm Tuesday 15 October

The main point of this lab is to gain experience with **parallel computing**. We will work with a simple CSV file and calculate test error for a simple linear regression model, exploring the time required for several different approaches.

The Data

The data set for this lab is a single large CSV file containing just the trip distance and trip duration variables for electric vehicle trips. This file is available on the VMs at the following location:

`/course/data.austintexas.gov/distance-duration.csv`

The Task

1. Read the complete CSV file using `read.csv()` and measure the execution time.
2. Use the shell command `split` to break the large CSV up into 10 roughly equal-size smaller CSV files (in your local directory).

   Hint: `man split` will give you a help page for the `split` command.
3. Read the 10 smaller CSV files into R in parallel and combine them to create a single data frame.

   Measure the execution time, compare that to the time taken to read the large CSV file, and check that this results in the same data frame.
4. Subset the data frame to remove non-positive trip distances and durations, and log both variables.
5. Using the following code ...

   ```
   labels <- rep(1:10, length.out=nrow(trips))
   groups <- sample(labels)
   mse <- function(i) {
       ## cat(i, "\n")
       testSet <- groups == i
       trainSet <- groups != i
       fit <- lm(logDuration ~ logDistance, trips, na.action=NULL)
       pred <- predict(fit, trips[testSet, ])
       mean((pred - trips$logDuration[testSet])^2, na.rm=TRUE)
   }
   ```

   ... estimate the test error by *k*-fold cross-validation as we have done in previous labs (we will call this the "serial" approach).

   Measure the execution time.
6. Repeat the test error estimate, but use `mclappy()` to perform the k-fold cross-validation in parallel. Check that this gives the same answer as the serial approach.

   Measure the execution time, compare that to the time taken by the serial approach.

7. Repeat the test error estimate again, but use `parLapply()` to perform the k-fold cross-validation in parallel. Check that this gives the same answer as the serial approach.

   Measure the execution time, compare that to the time taken by the serial approach and compare it to the time taken by the `mclapply()` approach.

8. Use the **caret** package (and the **doParallel** package) to estimate the test error *in parallel.*

   Hints: use the `doParallel::registerDoParallel()` function to set up parallel computation with **caret**; use the `caret::train()` function to estimate test error.

   ```
   train(formula, data= , method= ,
         trControl=trainControl(method= , number= ))
   ```

   Some potentially useful links:
   `https://topepo.github.io/caret/parallel-processing.html`
   `https://topepo.github.io/caret/train-models-by-tag.html#linear-regression`
   `https://topepo.github.io/caret/model-training-and-tuning.html#control`

   Measure the execution time, compare that to the time taken by the previous approaches, and check that this gives approximately the same answer as the serial approach.

The Report

Your submission should consist of a *tar ball* (as generated by the `tar` shell tool) that contains an R Markdown document *and* a Makefile *and* a processed version of your R Markdown document, submitted via Canvas.

You should write your document and your Makefile so that the tar ball can be extracted into a directory anywhere on one of the virtual machines provided for this course (`sc-cer00014-04.its.auckland.ac.nz` or `sc-cer00014-05.its.auckland.ac.nz`) and the markdown document can be processed just by typing `make`.

Your report should include:

- A description of the data format.
- A description of each of "The Task"s.
- A conclusion summarising the analysis.

Your report should NOT be longer than **10 pages**.

Your report should NOT take longer than **2 minutes** to process.

Your tar ball should NOT be larger than **1 Mb**.

Marks will be lost for:

- Submission is not a tar ball.
- More than 10 pages in the report.
- R Markdown file does not run.
- A "Task" is missing.
- R Markdown file is missing.
- Processed file (pdf or html) is missing.
- Makefile is missing.
- Significantly poor R (or other) code.