Portfolio 5: Parallel Rcpp

Kieran Morris

For this document we will be using the Irish smart meters dataset from the `electbook` package. We first import the data and display the first few rows.

Preprocessing

```
library(electBook)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(skimr)
data("Irish")
```

We concatonate the electricity data into one vector and center it:

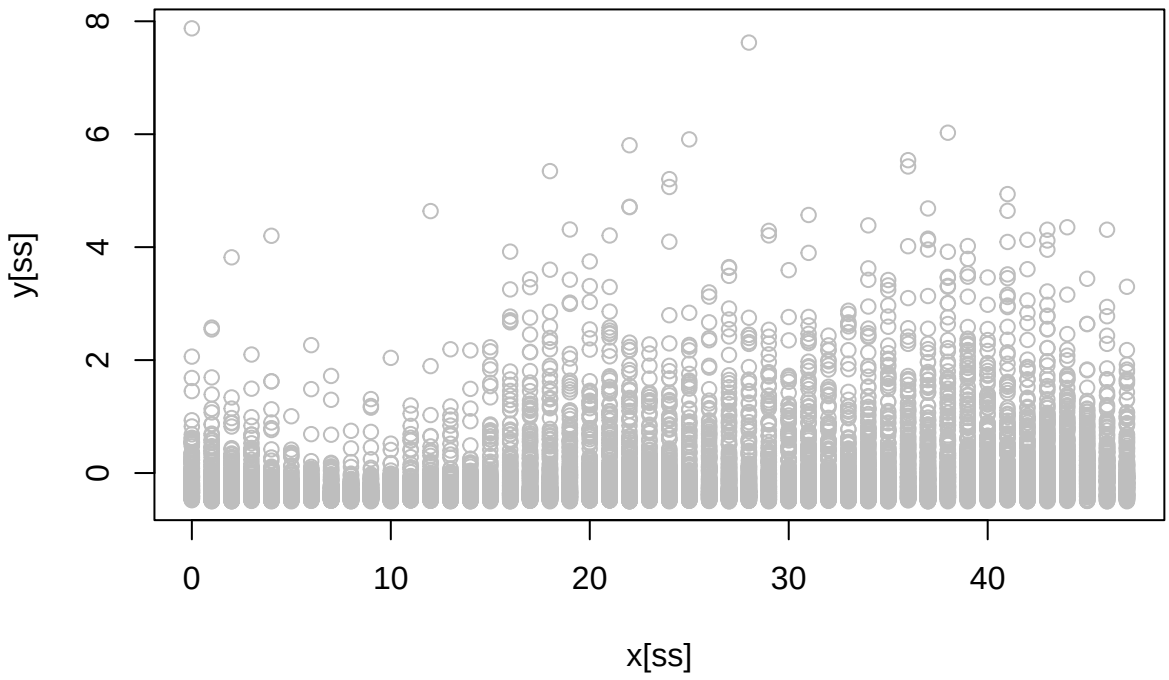
```
y <- do.call("c", Irish$indCons)
y <- y - mean(y)
```

and we plot it against time of day:

```
ncust <- ncol(Irish$indCons)

x <- rep(Irish$extra$tod, ncust)

n <- length(x)
ss <- sample(1:n, 1e4)
plot(x[ss], y[ss], col = "grey")
```



Simple Regression R

We will now fit a simple linear regression model to the data using the following function:

```
reg1D <- function(y, x){

  b <- t(x) %*% y / (t(x) %*% x)

  return(b)

}
```

This function actually performs very fast, although that's probably because it barely does any computation.

```
system.time( reg1D(y, x) )[3]
```

```
## elapsed
##   0.407
```

Simple Regression with RcppParallel

We will now implement the same function using RcppParallel. We will use inline Rcpp to do this.

```
Rcpp::sourceCpp(code='
// [[Rcpp::plugins(openmp)]]

#include <Rcpp.h>
#ifdef _OPENMP
#include <omp.h>
#endif

using namespace Rcpp;

// [[Rcpp::export]]
double parallelReg1D(NumericVector y, NumericVector x) {
   int n = x.size();
   double b = 0.0;
   double x_dot_x = 0.0;

   // Use OpenMP to perform the computation in parallel
   #pragma omp parallel for reduction(+:b, x_dot_x)
   for (int i = 0; i < n; i++) {
      b += x[i] * y[i];
      x_dot_x += x[i] * x[i];
   }

   return b / x_dot_x;
}
')
```

```
parallelReg1D(y, x)
```

```
## [1] 0.003250122
```

```
reg1D(y, x)
```

```
##             [,1]
## [1,] 0.003250122
```

As we can see they line up perfectly! We can also check the speed of the parallel function by using microbenchmark.

```
library(microbenchmark)

microbenchmark(
  "R" = reg1D(y, x),
  "Rccp" = parallelReg1D(y, x)
)
```

```
## Unit: milliseconds
##  expr       min        lq      mean    median       uq        max neval
##     R 405.67034 414.66840 569.94407 623.59312 698.8429 1391.98530   100
##  Rccp  11.00953  11.05726  11.53437  11.25532  11.4278   24.07174   100
```