

Random permutation of arrays

Statement of the problem

Our purpose is to

- study an algorithm for randomly permuting an array, and
- compare the run times of implementations in
 - **R**
 - **Rcpp**
 - the built-in `sample()` function in R.

The algorithm

PERMUTE(A)

```
n = length(A)
for (i = 1, ..., n) {
    SWAP(A, i, RANDOM(i, n))
}
```

Implementation

The following is an R implementation of this algorithm.

```
random <- function(a, b)
{
    a + floor(runif(1) * (b-a+1))
}
```

```
permute <- function(x)
```

```

{
  n <- length(x)
  for (i in seq_along(x))
  {
    s <- random(i, n)
    x[c(i, s)] <- x[c(s, i)]
  }
  x
}
permute(1:100)

```

One example of runtime:

```

> system.time(permute(1:1000000))
   user  system elapsed 
5.540    0.008    5.548 
> system.time(sample(1:1000000))
   user  system elapsed 
0.032    0.004    0.036 

```