Rcpp — Solutions to Exercises

Rcpp — Exercises Part 1

- Find out why the following code gives a compile error:

```
#include<Rcpp.h>
// [[Rcpp::plugins("cpp11")]]

NumericVector x{1, 2, 3, 4, 5};
IntegerVector id{1};

double y = x[id]; // produces compile error
```

Solution:

`x[id]` returns a subview class of `Rcpp::Vector` which is *not* a double. It is important that coercion to another type is not as easily done as in R!

- Benchmark the functions below against each other for `x<-rnorm(1e2)`, `x<-rnorm(1e4)` and `x<-rnorm(1e6)`. Comment on the results.

```
NumericVector test_clone_return(NumericVector A) {
  NumericVector B = clone(A);
  B[1] = 0.5;
  return B;
}
```

```
NumericVector test_reference_return(NumericVector A) {
  A[1] = 0.5;
  return A;
}
```

Solution:

```
bench::press(
  A = c(1e2, 1e4, 1e6),
  {
    A <- rnorm(A)
    bench::mark(
      test_clone_return(A),
      test_reference_return(A),
      check = F,
      relative = T
    )
  }
)
```

Obviously, cloning (that is copying on function call) is a bad idea!

Slide 52: `Rcpp` – STL Algorithms

The second appearence of `x.begin()` refers to the biginning of the output range, i.e. the first element of `x`.