

# Advanced R Programming - Lecture 6

## Rcpp

Krzysztof Bartoszek, Shashi Nagarajan  
(slides based on Leif Jonsson's and Måns Magnusson's)

Linköping University

*krzysztof.bartoszek@liu.se*

29 IX, 1 X 2025 (U2)

# Today

Rcpp

Memoization

# Questions since last time?

# Rcpp

Using C++ code in R

Need C++ compiler (look  
`http://adv-r.had.co.nz/Rcpp.html`)

Often called interfacing

Similar can be done with Java and Fortran

Extremely fast!

But just handle bottlenecks!

# Fibonacci

$$f(n) = \begin{cases} n, & \text{if } n < 2 \\ F(n-1) + F(n-2), & \text{otherwise} \end{cases}$$

# Fibonacci R

```
fr <- function(n) {  
  if (n < 2) return(n)  
  fr(n-1) + fr(n-2)  
}
```

```
system.time(fr(33))  
user      system elapsed  
3.312      0.008      3.32
```

# Fibonacci C++

```
library(Rcpp)

cppFunction(code = '
  int fcpp(int n) {
    if (n < 2) return(n);
    return(fcpp(n-1) + fcpp(n-2));
  }
',)

system.time(fcpp(33))
user      system elapsed
0.019      0.000      0.019
```

# Mapping R Datatypes to C++

R and C++ support different data types. The package Rcpp implements the RObject class for the following mappings

<u>R Datatype</u>	<u>Corresponding RObject Class</u>
integer scalar	int (no surprise)
logical scalar	bool (no surprise)
...	...
integer vector	IntegerVector
numeric vector	NumericVector
logical vector	LogicalVector
character vector	CharacterVector
...	...
numeric matrix	NumericMatrix
...	...

Chapters 3-4 of 'Seamless R and C++ Integration with Rcpp', Eddebuettel (2013) mention all such mappings



## Binary Search

- ▶ This example uses the NumericVector class to pass an R numeric vector as input to a C++ programme
- ▶ Idea: Search for an element in the vector using Binary Search
- ▶ The R and C++ implementations of the Binary Search algorithm available in `binary_search_rcpp.R`.

```
> microbenchmark(times=3000,
+               binary_search_R(sorted_x, 56, 1000),
+               binary_search_Cpp(sorted_x, 56, 1000))
Unit: microseconds
```

	expr	min	lq	mean	median	uq	max	neval	cld
	binary_search_R(sorted_x, 56, 1000)	2.101	2.2380	5.441608	2.318	2.414	9095.271	3000	a
	binary_search_Cpp(sorted_x, 56, 1000)	1.955	2.1035	2.657741	2.192	2.331	1046.548	3000	a

**Figure:** Performance Benchmarking Results

The example here was *not* designed to show excellent performance gains through Rcpp, but rather to show how to pass a vector to an Rcpp programme. A wealth of performant examples available at <https://gallery.rcpp.org/>

# Memoization

A simple optimization technique

Example of a general technique in optimization of trading memory  
for computation

Memoization stores (caches) results of function calls

If called again, returns old value

Depends on functional programming

Useful in recursive programming  
(instead of manual look-up structure)

## Memoise in R

```
> library(memoise)
> a <- function(x) runif(1)
> replicate(3, a())
[1] 0.6709919 0.3490709 0.4772027
> b <- memoise(a)
> replicate(3, b())
[1] 0.1867441 0.1867441 0.1867441
```

## Memoise in R

```
> c <- memoise(function(x) {Sys.sleep(1); runif(1)})  
> system.time(print(c()))  
[1] 0.7816399  
user    system elapsed  
0.003    0.004    1.001  
> system.time(print(c()))  
[1] 0.7816399  
user    system elapsed  
0.001    0.000    0.000  
> forget(c)  
[1] TRUE  
> system.time(print(c()))  
[1] 0.9234995  
user    system elapsed  
0.003    0.004    1.001
```

# Memoise Fibonacci

```
library(memoise)

frm <- memoise(function(n) {
  if (n < 2) return(n)
  frm(n-1) + frm(n-2)
})

system.time(frm(33))
user      system elapsed
0.029      0.000      0.029
```

Memoisation of C++ code did not improve timing.  
Memoisation can be sometimes an alternative to C++.

<https://www.inwt-statistics.com/read-blog/optimize-your-r-code-using-memoization.html>

The End... for today.  
Questions?  
See you next time!