

Contents

Part I Introduction

| | | |
|----------|---|-----------|
| 1 | A Gentle Introduction to Rcpp | 3 |
| 1.1 | Background: From R to C++ | 3 |
| 1.2 | A First Example | 7 |
| 1.2.1 | Problem Setting | 7 |
| 1.2.2 | A First R Solution | 7 |
| 1.2.3 | A First C++ Solution | 8 |
| 1.2.4 | Using Inline | 9 |
| 1.2.5 | Using Rcpp Attributes | 11 |
| 1.2.6 | A Second R Solution | 12 |
| 1.2.7 | A Second C++ Solution | 12 |
| 1.2.8 | A Third R Solution | 14 |
| 1.2.9 | A Third C++ Solution | 14 |
| 1.3 | A Second Example | 15 |
| 1.3.1 | Problem Setting | 15 |
| 1.3.2 | R Solution | 15 |
| 1.3.3 | C++ Solution | 16 |
| 1.3.4 | Comparison | 17 |
| 1.4 | Summary | 18 |
| 2 | Tools and Setup | 19 |
| 2.1 | Overall Setup | 19 |
| 2.2 | Compilers | 20 |
| 2.2.1 | General Setup | 20 |
| 2.2.2 | Platform-Specific Notes | 21 |
| 2.3 | The R Application Programming Interface | 22 |
| 2.4 | A First Compilation with Rcpp | 23 |
| 2.5 | The Inline Package | 25 |
| 2.5.1 | Overview | 25 |
| 2.5.2 | Using Includes | 27 |

| | | |
|-------|--------------------|----|
| 2.5.3 | Using Plugins | 29 |
| 2.5.4 | Creating Plugins | 30 |
| 2.6 | Rcpp Attributes | 31 |
| 2.7 | Exception Handling | 32 |

Part II Core Data Types

| | | |
|----------|--|-----------|
| 3 | Data Structures: Part One | 39 |
| 3.1 | The RObject Class | 39 |
| 3.2 | The IntegerVector Class | 41 |
| 3.2.1 | A First Example: Returning Perfect Numbers | 42 |
| 3.2.2 | A Second Example: Using Inputs | 43 |
| 3.2.3 | A Third Example: Using Wrong Inputs | 44 |
| 3.3 | The NumericVector Class | 45 |
| 3.3.1 | A First Example: Using Two Inputs | 45 |
| 3.3.2 | A Second Example: Introducing clone | 46 |
| 3.3.3 | A Third Example: Matrices | 47 |
| 3.4 | Other Vector Classes | 48 |
| 3.4.1 | LogicalVector | 48 |
| 3.4.2 | CharacterVector | 49 |
| 3.4.3 | RawVector | 49 |
| 4 | Data Structures: Part Two | 51 |
| 4.1 | The Named Class | 51 |
| 4.2 | The List aka GenericVector Class | 52 |
| 4.2.1 | List to Retrieve Parameters from R | 53 |
| 4.2.2 | List to Return Parameters to R | 54 |
| 4.3 | The DataFrame Class | 55 |
| 4.4 | The Function Class | 56 |
| 4.4.1 | A First Example: Using a Supplied Function | 56 |
| 4.4.2 | A Second Example: Accessing an R Function | 56 |
| 4.5 | The Environment Class | 57 |
| 4.6 | The S4 Class | 58 |
| 4.7 | ReferenceClasses | 59 |
| 4.8 | The R Mathematics Library Functions | 60 |

Part III Advanced Topics

| | | |
|----------|-----------------------------------|-----------|
| 5 | Using Rcpp in Your Package | 65 |
| 5.1 | Introduction | 65 |
| 5.2 | Using Rcpp.package.skeleton | 66 |
| 5.2.1 | Overview | 66 |
| 5.2.2 | R Code | 67 |
| 5.2.3 | C++ Code | 68 |
| 5.2.4 | DESCRIPTION | 69 |
| 5.2.5 | Makevars and Makevars.win | 69 |

| | | |
|----------|---|-----|
| 5.2.6 | NAMESPACE | 71 |
| 5.2.7 | Help Files | 71 |
| 5.3 | Case Study: The wordcloud Package | 73 |
| 5.4 | Further Examples | 74 |
| 6 | Extending Rcpp | 75 |
| 6.1 | Introduction | 75 |
| 6.2 | Extending Rcpp::wrap | 76 |
| 6.2.1 | Intrusive Extension | 76 |
| 6.2.2 | Nonintrusive Extension | 77 |
| 6.2.3 | Templates and Partial Specialization | 78 |
| 6.3 | Extending Rcpp::as | 78 |
| 6.3.1 | Intrusive Extension | 78 |
| 6.3.2 | Nonintrusive Extension | 79 |
| 6.3.3 | Templates and Partial Specialization | 79 |
| 6.4 | Case Study: The RcppBDT Package | 80 |
| 6.5 | Further Examples | 82 |
| 7 | Modules | 83 |
| 7.1 | Motivation | 83 |
| 7.1.1 | Exposing Functions Using Rcpp | 83 |
| 7.1.2 | Exposing Classes Using Rcpp | 84 |
| 7.2 | Rcpp Modules | 86 |
| 7.2.1 | Exposing C++ Functions Using Rcpp Modules | 86 |
| 7.2.2 | Exposing C++ Classes Using Rcpp Modules | 90 |
| 7.3 | Using Modules in Other Packages | 98 |
| 7.3.1 | Namespace Import/Export | 98 |
| 7.3.2 | Support for Modules in Skeleton Generator | 99 |
| 7.3.3 | Module Documentation | 100 |
| 7.4 | Case Study: The RcppCNPY Package | 100 |
| 7.5 | Further Examples | 102 |
| 8 | Sugar | 103 |
| 8.1 | Motivation | 103 |
| 8.2 | Operators | 105 |
| 8.2.1 | Binary Arithmetic Operators | 105 |
| 8.2.2 | Binary Logical Operators | 106 |
| 8.2.3 | Unary Operators | 106 |
| 8.3 | Functions | 107 |
| 8.3.1 | Functions Producing a Single Logical Result | 107 |
| 8.3.2 | Functions Producing Sugar Expressions | 107 |
| 8.3.3 | Mathematical Functions | 113 |
| 8.3.4 | The d/q/p/q Statistical Functions | 114 |
| 8.4 | Performance | 115 |
| 8.5 | Implementation | 116 |

| | | |
|-------|---|-----|
| 8.5.1 | The Curiously Recurring Template Pattern | 117 |
| 8.5.2 | The VectorBase Class | 117 |
| 8.5.3 | Example: supply | 118 |
| 8.6 | Case Study: Computing π Using <i>Rcpp sugar</i> | 122 |

Part IV Applications

| | | |
|-----------|---|-----|
| 9 | RInside | 127 |
| 9.1 | Motivation | 127 |
| 9.2 | A First Example: Hello, World! | 128 |
| 9.3 | A Second Example: Data Transfer | 131 |
| 9.4 | A Third Example: Evaluating R Expressions | 132 |
| 9.5 | A Fourth Example: Plotting from C++ via R | 133 |
| 9.6 | A Fifth Example: Using RInside Inside MPI | 134 |
| 9.7 | Other Examples | 135 |
| 10 | RcppArmadillo | 139 |
| 10.1 | Overview | 139 |
| 10.2 | Motivation: FastLm | 140 |
| 10.2.1 | Implementation | 140 |
| 10.2.2 | Performance Comparison | 142 |
| 10.2.3 | A Caveat | 144 |
| 10.3 | Case Study: Kalman Filter Using RcppArmadillo | 146 |
| 10.4 | RcppArmadillo and Armadillo Differences | 152 |
| 11 | RcppGSL | 155 |
| 11.1 | Introduction | 155 |
| 11.2 | Motivation: FastLm | 156 |
| 11.3 | Vectors | 158 |
| 11.3.1 | GSL Vectors | 158 |
| 11.3.2 | RcppGSL::vector | 159 |
| 11.3.3 | Mapping | 161 |
| 11.3.4 | Vector Views | 161 |
| 11.4 | Matrices | 163 |
| 11.4.1 | Creating Matrices | 163 |
| 11.4.2 | Implicit Conversion | 163 |
| 11.4.3 | Indexing | 163 |
| 11.4.4 | Methods | 164 |
| 11.4.5 | Matrix Views | 164 |
| 11.5 | Using RcppGSL in Your Package | 164 |
| 11.5.1 | The <code>configure</code> Script | 165 |
| 11.5.2 | The <code>src</code> Directory | 166 |
| 11.5.3 | The <code>R</code> Directory | 167 |
| 11.6 | Using RcppGSL with inline | 168 |
| 11.7 | Case Study: GSL -Based B-Spline Fit Using RcppGSL | 169 |

| | |
|--|-----|
| 12 RcppEigen | 177 |
| 12.1 Introduction | 177 |
| 12.2 Eigen classes | 178 |
| 12.2.1 Fixed-Size Vectors and Matrices | 178 |
| 12.2.2 Dynamic-Size Vectors and Matrices | 179 |
| 12.2.3 Arrays for Per-Component Operations | 180 |
| 12.2.4 Mapped Vectors and Matrices and Special Matrices | 181 |
| 12.3 Case Study: Kalman filter using RcppEigen | 182 |
| 12.4 Linear Algebra and Matrix Decompositions | 183 |
| 12.4.1 Basic Solvers | 183 |
| 12.4.2 Eigenvalues and Eigenvectors | 184 |
| 12.4.3 Least-Squares Solvers | 185 |
| 12.4.4 Rank-Revealing Decompositions | 185 |
| 12.5 Case Study: C++ Factory for Linear Models in RcppEigen | 186 |

Part V Appendix

| | |
|---|-----|
| A C++ for R Programmers | 195 |
| A.1 Compiled Not Interpreted | 195 |
| A.2 Statically Typed | 197 |
| A.3 A Better C | 198 |
| A.4 Object-Oriented (But Not Like S3 or S4) | 200 |
| A.5 Generic Programming and the STL | 201 |
| A.6 Template Programming | 203 |
| A.7 Further Reading on C++ | 204 |
| References | 207 |
| Subject Index | 211 |
| Software Index | 217 |
| Author Index | 219 |

<http://www.springer.com/978-1-4614-6867-7>

Seamless R and C++ Integration with Rcpp

Eddelbuettel, D.

2013, XXVIII, 220 p. 7 illus., 4 illus. in color., Softcover

ISBN: 978-1-4614-6867-7