R as an interface to C++: exchanging convenience for speed

<May, 2018>

https://github.com/pgurazada

Who am I?

- Sales guy
- Right of the middle in $\boldsymbol{y} = f(\boldsymbol{X})$

Agenda

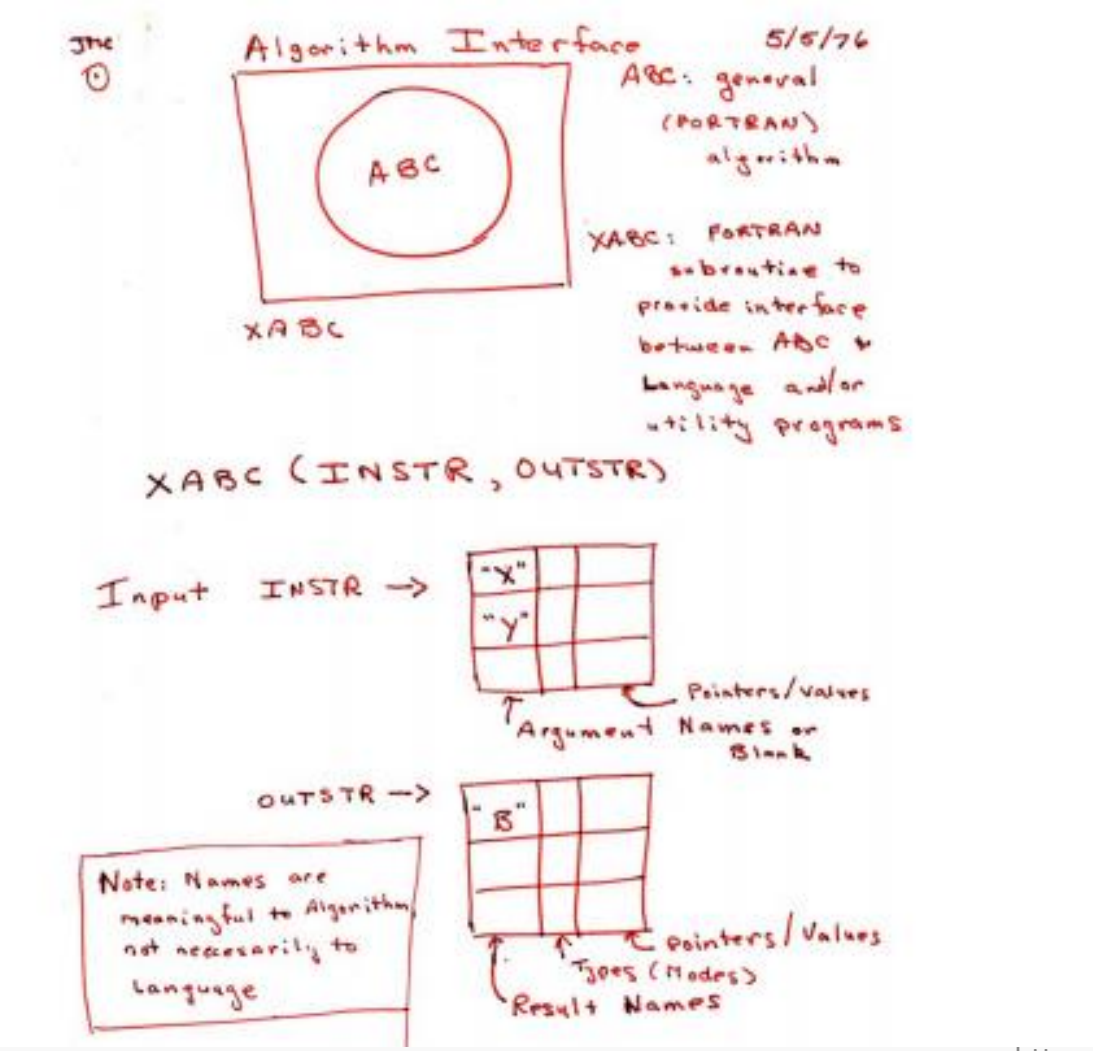
- R is an interface
- R might be fast
- Mapping R objects to C++ with Rcpp
- (Interesting) Examples
- Starting today…

This is a highly opinionated talk…

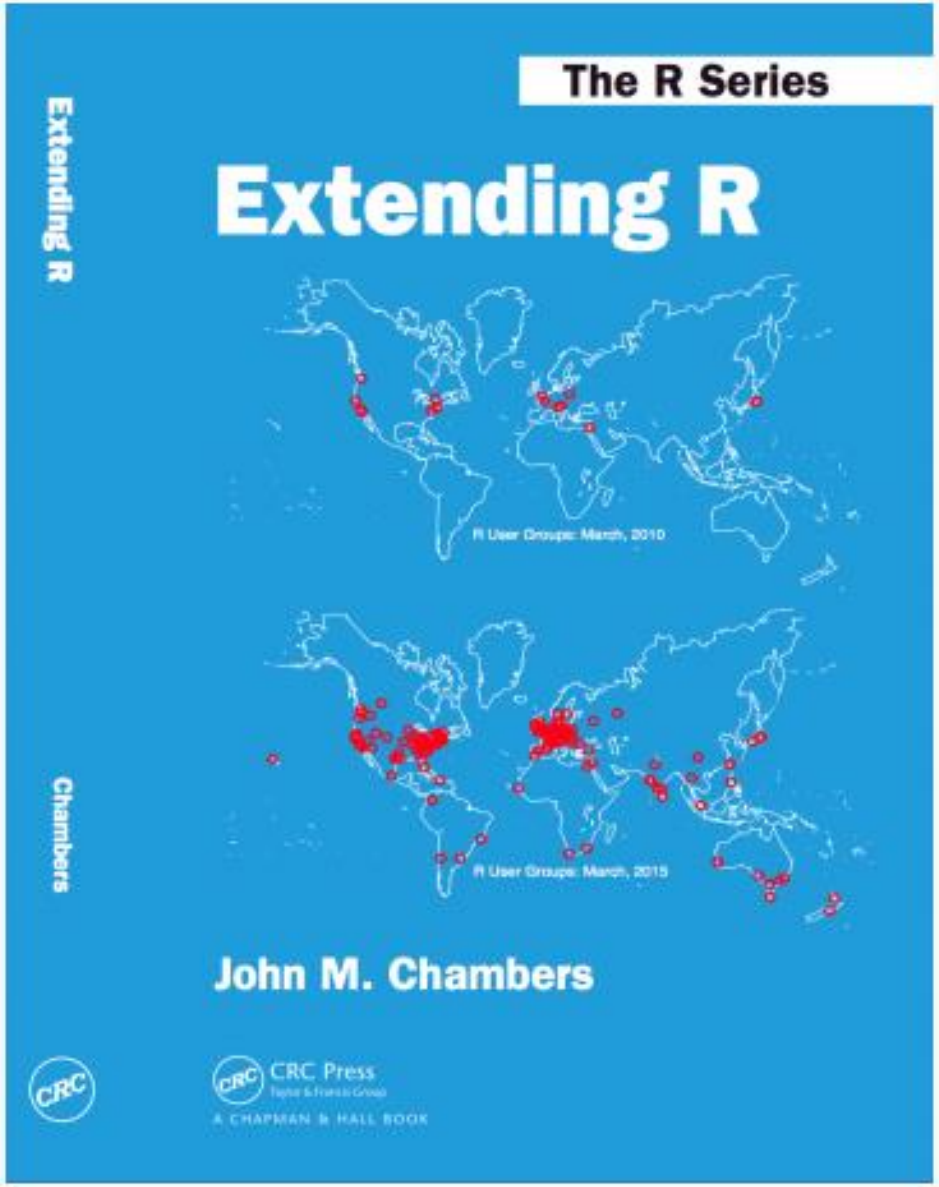
Agenda

- R is an interface
- R might be fast
- Mapping R objects to C++ with Rcpp
- (Interesting) Examples
- Starting today…

When S came before R…



https://github.com/pgurazada



johnmchambers / **XRJulia**

Watch 4 | Star 24 | Fork 1

Code | Issues 5 | Pull requests 0 | Projects 0 | Wiki | Insights

XR-style Interface to Julia (from "Extending R")

R 68.3% | Julia 31.7%

johnmchambers / **XRPython**

Watch 2 | Star 13 | Fork 1

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights

XR-style Interface to Python (from "Extending R")

R 83.4% | Python 16.6%

https://github.com/pgurazada

Andreas Mueller Retweeted

Wes McKinney
@wesmckinn

Big news today! I've founded Ursa Labs, a development lab for open source data science, powered by @ApacheArrow. I'm teaming up with @hadleywickham and @rstudio to make it possible wesmckinney.com/blog/announcin...

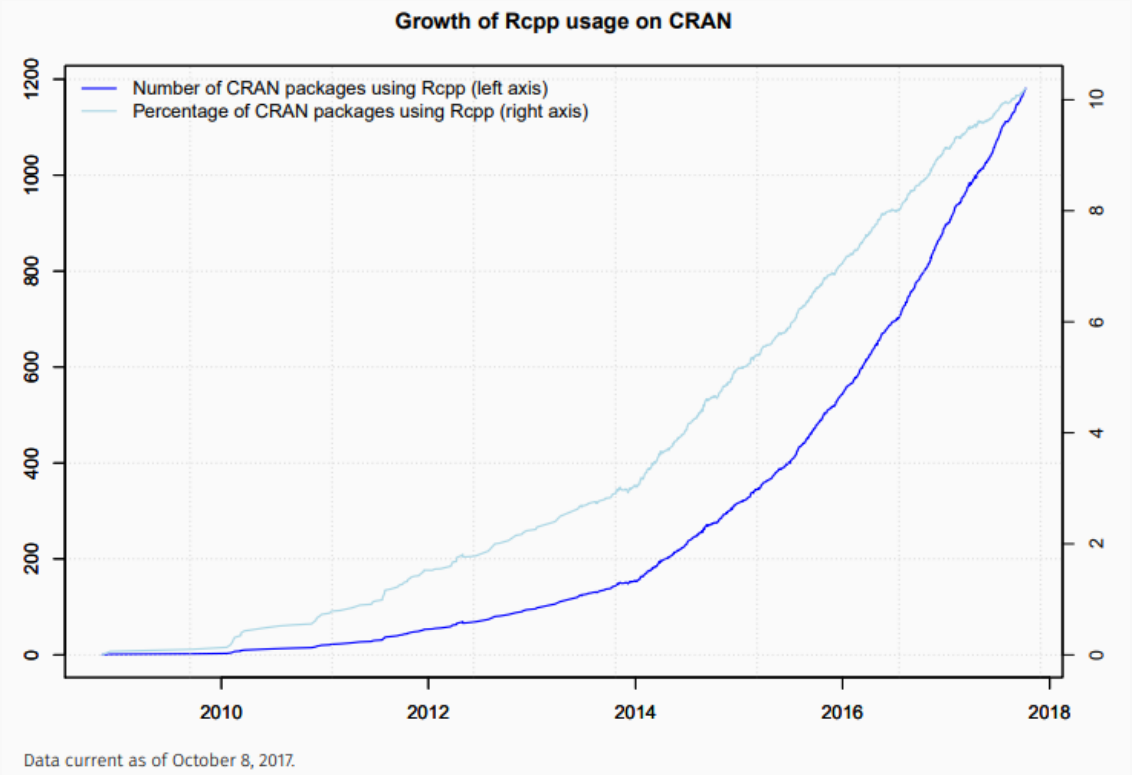
4/19/18, 8:00 PM

Rcpp

build passing | license GPL (>= 2) | CRAN 0.12.16 | downloads 689K/month | codecov 93%

Seamless R and C++ Integration

The Rcpp package provides R functions and a C++ library facilitating the integration of R and C++.



https://github.com/pgurazada

Agenda

- R is an interface
- R might be fast
- Mapping R objects to C++ with Rcpp
- (Interesting) Examples
- Starting today…

Look before you leap…

- Use R 3.4+ (reason: bytecode)
- R 3.5 just landed; upgrade (reason: ALTREP)
- Vectorize (reason: everyone is doing it)

Agenda

- R is an interface
- R might be fast
- Mapping R objects to C++ with Rcpp
- (Interesting) Examples
- Starting today…

Start thinking about types

Value	R vector	Rcpp vector	Rcpp matrix	Rcpp scalar	C++ scalar
Logical	`logical`	`LogicalVector`	`LogicalMatrix`	-	`bool`
Integer	`integer`	`IntegerVector`	`IntegerMatrix`	-	`int`
Real	`numeric`	`NumericVector`	`NumericMatrix`	-	`double`
Complex	`complex`	`ComplexVector`	`ComplexMatrix`	`Rcomplex`	`complex`
String	`character`	`CharacterVector` (`StringVector`)	`CharacterMatrix` (`StringMatrix`)	`String`	`string`
Date	`Date`	`DateVector`	-	`Date`	-
Datetime	`POSIXct`	`DatetimeVector`	-	`Datetime`	`time_t`

(Masaki, 2018)

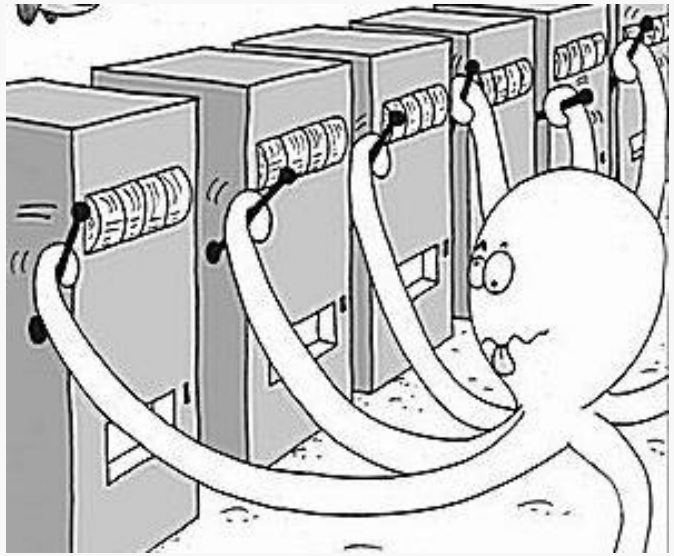
Agenda

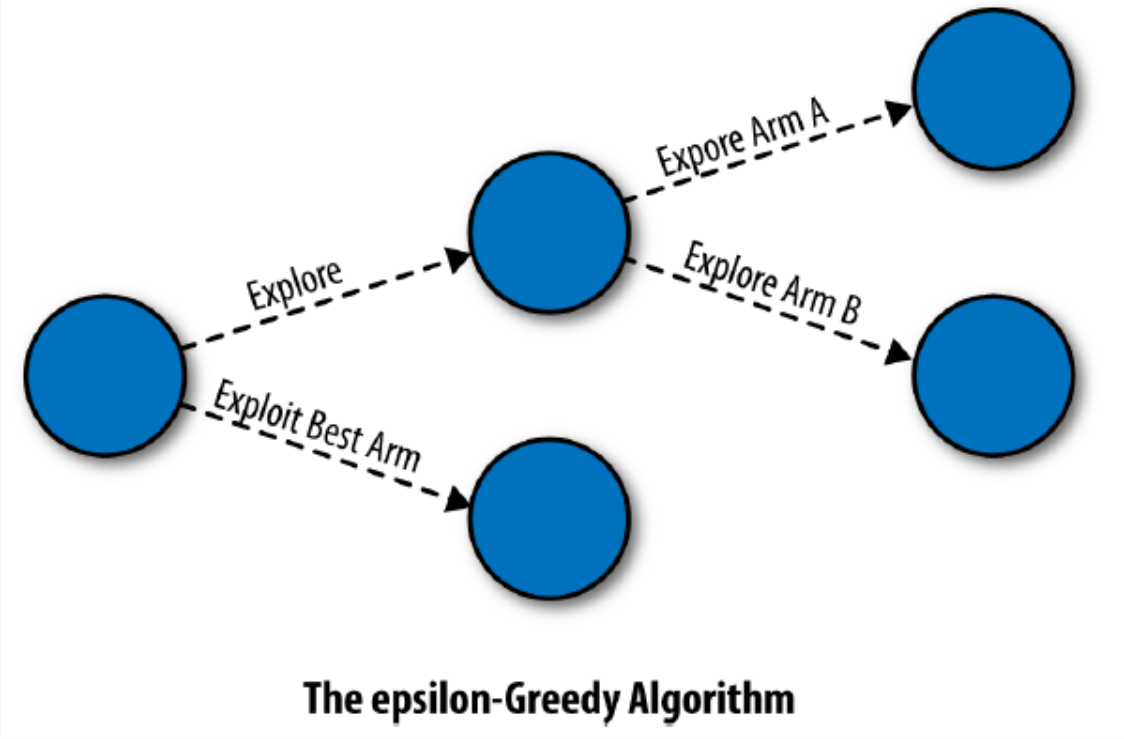
- R is an interface
- R might be fast
- Mapping R objects to C++ with Rcpp
- (Interesting) Examples
- Starting today…

A typical workflow

- Write R functions for repetitive tasks
- Profile if slow
- Note types
- Translate it into Rcpp

Implementing multi-armed bandits





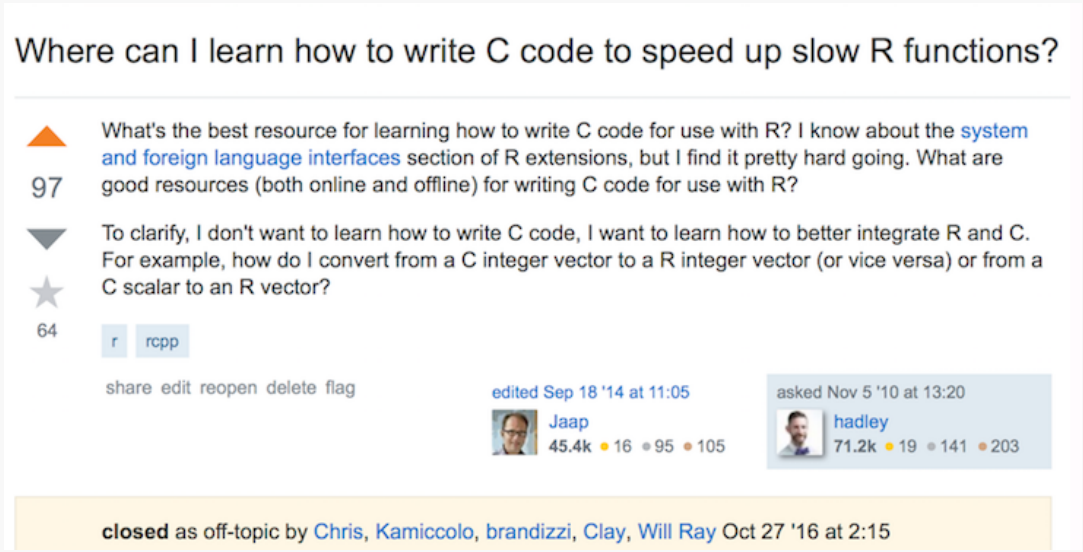
The epsilon-Greedy Algorithm

(John Myles White, 2013)

Agenda

- R is an interface
- R might be fast
- Mapping R objects to C++ with Rcpp
- (Interesting) Examples
- Starting today…

If you are starting out today…



Where can I learn how to write C code to speed up slow R functions?

97

What's the best resource for learning how to write C code for use with R? I know about the system and foreign language interfaces section of R extensions, but I find it pretty hard going. What are good resources (both online and offline) for writing C code for use with R?

To clarify, I don't want to learn how to write C code, I want to learn how to better integrate R and C. For example, how do I convert from a C integer vector to a R integer vector (or vice versa) or from a C scalar to an R vector?

64

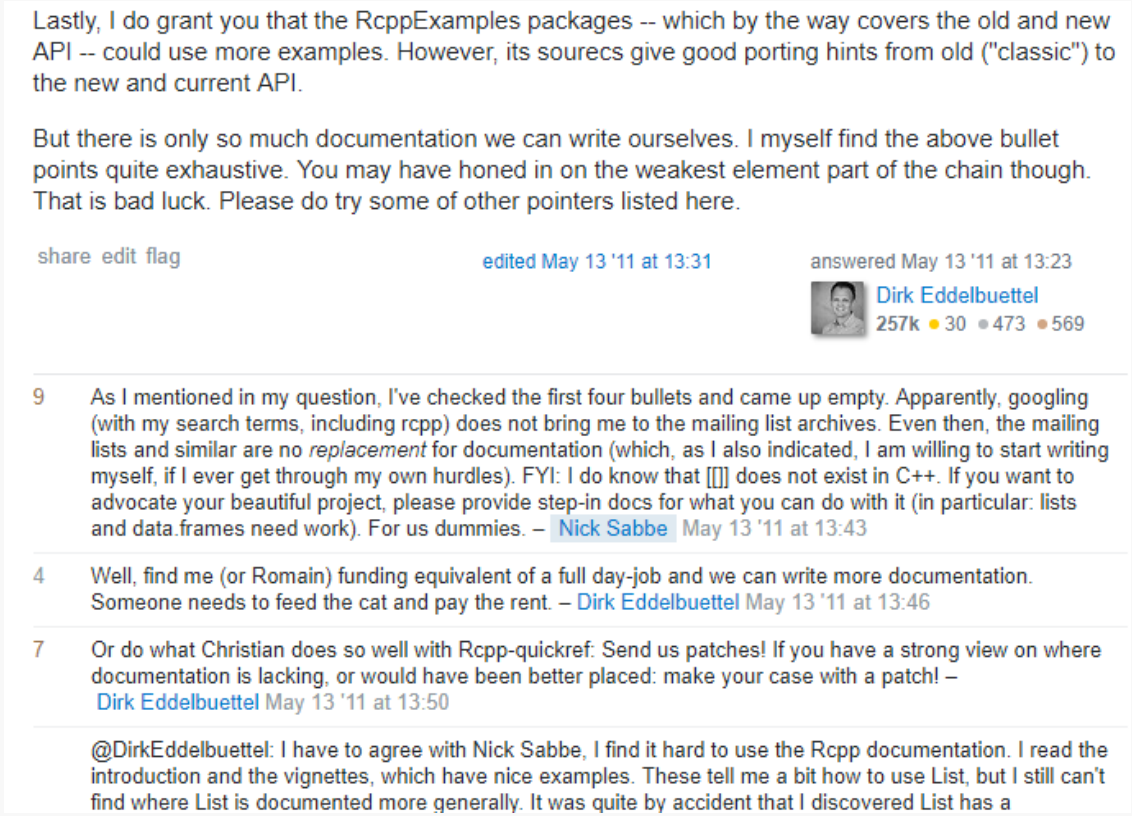
r rcpp

share edit reopen delete flag

edited Sep 18 '14 at 11:05
Jaap
45.4k • 16 • 95 • 105

asked Nov 5 '10 at 13:20
hadley
71.2k • 19 • 141 • 203

closed as off-topic by Chris, Kamiccolo, brandizzi, Clay, Will Ray Oct 27 '16 at 2:15



Lastly, I do grant you that the RcppExamples packages -- which by the way covers the old and new API -- could use more examples. However, its sourecs give good porting hints from old ("classic") to the new and current API.

But there is only so much documentation we can write ourselves. I myself find the above bullet points quite exhaustive. You may have honed in on the weakest element part of the chain though. That is bad luck. Please do try some of other pointers listed here.

share edit flag

edited May 13 '11 at 13:31

answered May 13 '11 at 13:23
Dirk Eddelbuettel
257k • 30 • 473 • 569

9 As I mentioned in my question, I've checked the first four bullets and came up empty. Apparently, googling (with my search terms, including rcpp) does not bring me to the mailing list archives. Even then, the mailing lists and similar are no *replacement* for documentation (which, as I also indicated, I am willing to start writing myself, if I ever get through my own hurdles). FYI: I do know that [[]] does not exist in C++. If you want to advocate your beautiful project, please provide step-in docs for what you can do with it (in particular: lists and data.frames need work). For us dummies. – Nick Sabbe May 13 '11 at 13:43

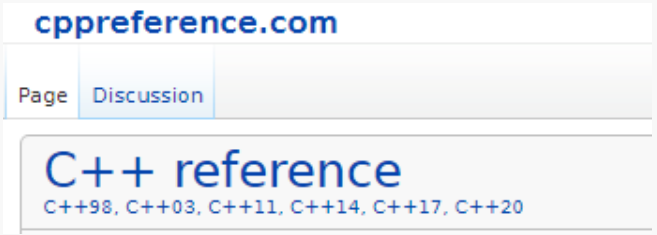
4 Well, find me (or Romain) funding equivalent of a full day-job and we can write more documentation. Someone needs to feed the cat and pay the rent. – Dirk Eddelbuettel May 13 '11 at 13:46

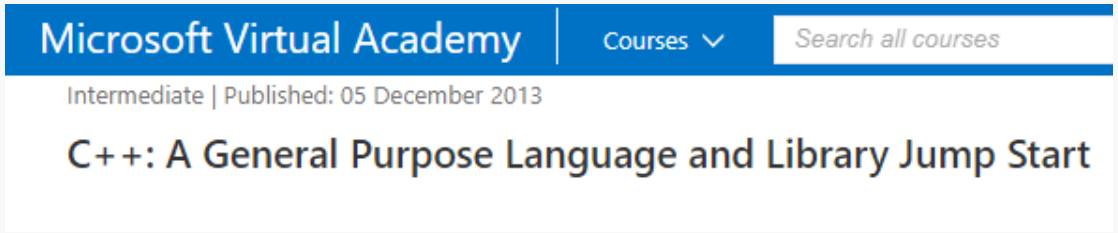
7 Or do what Christian does so well with Rcpp-quickref: Send us patches! If you have a strong view on where documentation is lacking, or would have been better placed: make your case with a patch! – Dirk Eddelbuettel May 13 '11 at 13:50

@DirkEddelbuettel: I have to agree with Nick Sabbe, I find it hard to use the Rcpp documentation. I read the introduction and the vignettes, which have nice examples. These tell me a bit how to use List, but I still can't find where List is documented more generally. It was quite by accident that I discovered List has a

https://github.com/pgurazada

If you are starting out today…





Questions?