R for Machine Learning

An Brief Overview with Examples

Dirk Eddelbuettel

6th COST Conference on AI in Industry and Finance

9 September 2021, Zurich University of Applied Sciences (ZHAW), CH

https://dirk.eddelbuettel.com/papers/zhaw_cost_sep2021.pdf

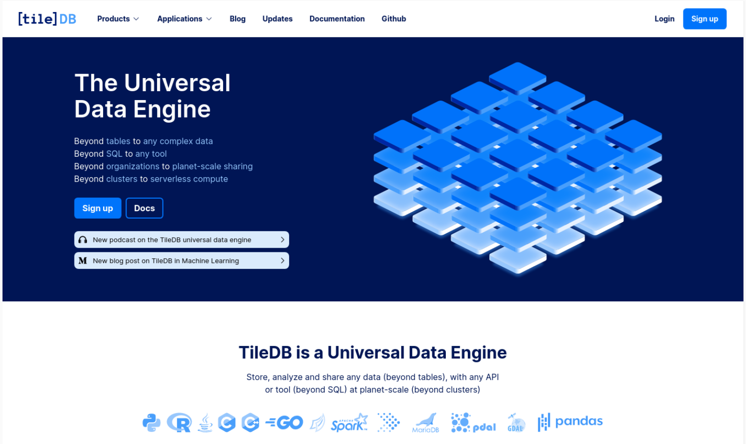
ILLINOIS

Outline

Today's Talk

- R as "interface"
- Rcpp as glue
- Machine Learning *Interfacing* Examples

Quick Bio: My day job



The Universal Data Engine

Beyond tables to any complex data
Beyond SQL to any tool
Beyond organizations to planet-scale sharing
Beyond clusters to serverless compute

New podcast on the TileDB universal data engine

New blog post on TileDB in Machine Learning

TileDB is a Universal Data Engine

Store, analyze and share any data (beyond tables), with any API or tool (beyond SQL) at planet-scale (beyond clusters)

Serverless, Planet-scale

Any data, any size

Local, or multi-cloud

Via an API into many languages and applications

Talk to us

Yes, we’re hiring

Quick Bio: My other jobs

Academic

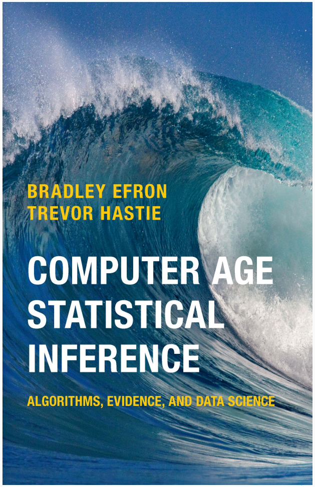
- (Adjunct) Clinical Professor, University of Illinois
  - teaching STAT 447, a Data Science Programming Methods class

Open Source

- Debian developer
  - since 1995, currently maintaining about 170+ packages
- R contributor / package author
  - since 2002, author / maintainer of 60+ CRAN packages, R Foundation Board Member
- Rocker Project co-founder
  - Docker for R, including official ‘r-base’ image

Why R: View from Academia

Computer-Age Statistical Inference


BRADLEY EFRON
TREVOR HASTIE
COMPUTER AGE STATISTICAL INFERENCE
ALGORITHMS, EVIDENCE, AND DATA SCIENCE

Almost all topics in twenty-first-century statistics are now computer-dependent [...]

Here and in all our examples we are employing the language R, itself one of the key developments in computer-based statistical methodology.

Efron and Hastie, 2016

pages xv and 6 (footnote 3)

Another View: James et al, ISLR (2nd ed)

> *[...] we have devoted a section within each chapter to R computer labs. In each lab, we walk the reader through a realistic application of the methods considered in that chapter. [...] We have used R because it is freely available and is powerful enough to implement all of the methods discussed in the book. It also has optional packages that can be downloaded to implement literally thousands of additional methods. Most importantly, R is the language of choice for academic statisticians, and new approaches often become available in R years before they are implemented in commercial packages.*

Another View: James et al, ISLR (2nd ed)

[...] we have devoted a section within each chapter to R computer labs. In each lab, we walk the reader through a realistic application of the methods considered in that chapter. [...] We have used R because it is freely available and is powerful enough to implement all of the methods discussed in the book. It also has optional packages that can be downloaded to implement literally thousands of additional methods. Most importantly, R is the language of choice for academic statisticians, and new approaches often become available in R years before they are implemented in commercial packages.

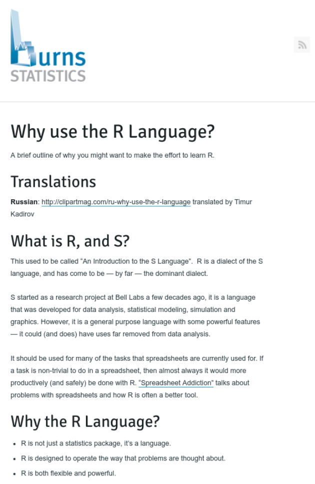
A View of the World

Computational Statistics in Practice

- Statistics is now computational (Efron & Hastie, 2016)
- Within (computational) statistics, reigning tool is R (James *et al*, ISLR, 2021)
- Given R, Rcpp key for two angles:
    - *Performance* always matters, ease of use a sweetspot
    - *“Extending R”* (Chambers, 2016)

Why R: View from Practitioners

Why R? Pat Burn's View



Why use the R Language?

A brief outline of why you might want to make the effort to learn R.

Translations

Russian: http://clipartmag.com/ru-why-use-the-r-language translated by Timur Kadirov

What is R, and S?

This used to be called "An Introduction to the S Language". R is a dialect of the S language, and has come to be — by far — the dominant dialect.

S started as a research project at Bell Labs a few decades ago, it is a language that was developed for data analysis, statistical modeling, simulation and graphics. However, it is a general purpose language with some powerful features — it could (and does) have uses far removed from data analysis.

It should be used for many of the tasks that spreadsheets are currently used for. If a task is non-trivial to do in a spreadsheet, then almost always it would more productively (and safely) be done with R. "Spreadsheet Addiction" talks about problems with spreadsheets and how R is often a better tool.

Why the R Language?

- R is not just a statistics package, it's a language.
- R is designed to operate the way that problems are thought about.
- R is both flexible and powerful.

Why the R Language?

Screen shot on the left part of short essay at Burns-Stat

His site has more truly excellent (and free) writings.

The (much longer) R Inferno (free pdf, also paperback) is highly recommended.

Why R? Pat Burn's View

Why the R Language?

- R is not just a statistics package, it's a language.
- R is designed to operate the way that problems are thought about.
- R is both flexible and powerful.

Source: https://www.burns-stat.com/documents/tutorials/why-use-the-r-language/

Why R? Pat Burn's View

Why R for data analysis?

R is not the only language that can be used for data analysis. Why R rather than another? Here is a list:

- interactive language
- data structures
- graphics
- missing values
- functions as first class objects
- packages
- community

Source: https://www.burns-stat.com/documents/tutorials/why-use-the-r-language/

Why R: Programming with Data

Why R? R as Middle Man

R as an Extensible Environment

- As R users we know that R can
  - ingest data in many formats from many sources
  - aggregate, slice, dice, summarize, ...
  - visualize in many forms, ...
  - model in just about any way
  - report in many useful and scriptable forms
- It has become central for programming with data
- Sometimes we want to extend it further than R code goes

R as central point

R AS CENTRAL POINT

From any one of

- csv
- txt
- xlsx
- xml, json, ...
- web scraping, ...
- hdf5, netcdf, ...
- sas, stata, spss, ...
- various SQL + NOSQL DBs
- various binary protocols

via

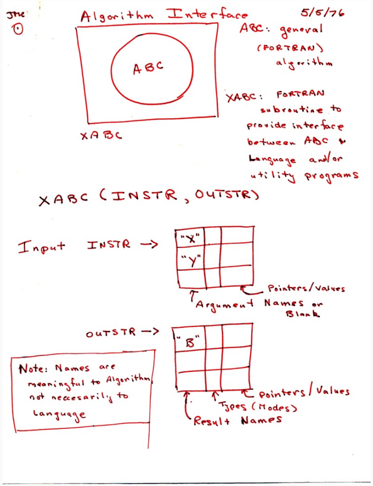
R

into any one of

- txt
- html
- latex and pdf
- html and js
- word
- shiny
- most graphics formats
- other dashboards
- web frontends

Why R: Historical Perspective

R as ‘The Interface’



JMC
①

Algorithm Interface 5/5/76

ABC: general (FORTRAN) algorithm

XABC: FORTRAN subroutine to provide interface between ABC & Language and/or utility programs

XABC (INSTR, OUTSTR)

Note: Names are meaningful to Algorithm, not neccesarily to language

A design sketch called ‘The Interface’

AT&T Research lab meeting notes

Describes an outer ‘user interface’ layer to core Fortran algorithms

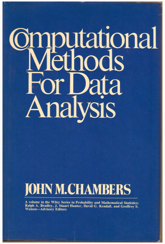
Key idea of abstracting away inner details giving higher-level more accessible view for user / analyst

Lead to “The Interface”

Which became S which lead to R

Source: John Chambers, personal communication

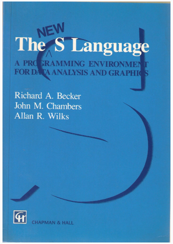
Why R? : *Programming with Data* from 1977 to 2016



Computational Methods For Data Analysis

JOHN M. CHAMBERS

A volume in the Wiley Series in Probability and Mathematical Statistics: Ralph A. Bradley, J. Stuart Hunter, David G. Kendall, and Geoffrey S. Watson—Advisory Editors

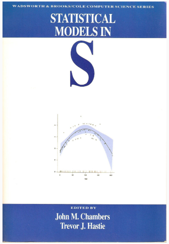


The NEW S Language

A PROGRAMMING ENVIRONMENT FOR DATA ANALYSIS AND GRAPHICS

Richard A. Becker
John M. Chambers
Allan R. Wilks

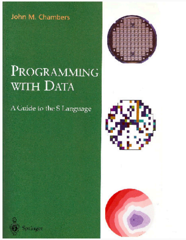
CHAPMAN & HALL



WADSWORTH & BROOKS/COLE COMPUTER SCIENCE SERIES

STATISTICAL MODELS IN S

EDITED BY
John M. Chambers
Trevor J. Hastie

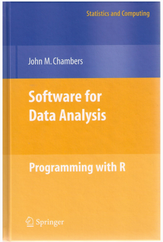


John M. Chambers

Programming with Data

A Guide to the S Language

Springer



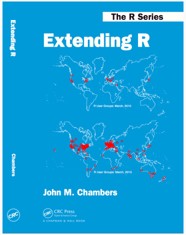
Statistics and Computing

John M. Chambers

Software for Data Analysis

Programming with R

Springer



The R Series

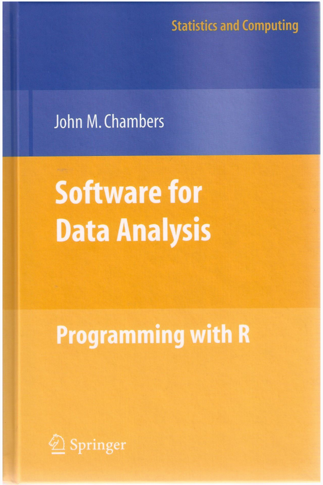
Extending R

John M. Chambers

CRC Press
A CHAPMAN & HALL BOOK

Thanks to John Chambers for high-resolution cover images. The publication years are, respectively, 1977, 1988, 1992, 1998, 2008 and 2016.

Chambers (2008)



Statistics and Computing

John M. Chambers

Software for
Data Analysis

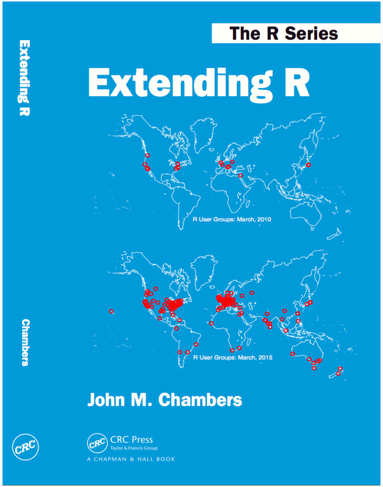
Programming with R

Springer

Software For Data Analysis

Chapters 10 and 11 devoted to *Interfaces I: C and Fortran* and *Interfaces II: Other Systems*.

Chambers (2016)



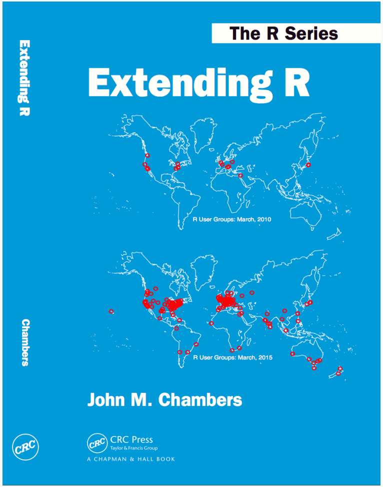
Extending R

Object: Everything that exists in R is an object

Function: Everything happens in R is a function call

Interface: Interfaces to other software are part of R

Chambers (2016)



Extending R, Chapter 4

The fundamental lesson about programming in the large is that requires a correspondingly broad and flexible response. In particular, no single language or software system os likely to be ideal for all aspects. Interfacing multiple systems is the essence. Part IV explores the design of of interfaces from R.

So ...

From a system called ‘interface’ to a language where interfaces are a natural part

Why Rcpp?

R and C/C++

A good fit, it turns out

- A good part of R is written in C (besides R and Fortran code)
- The principle interface to external code is a function `.Call()`
- It takes one or more of the high-level data structures R uses
- … and returns one. Formally:

```
SEXP .Call(SEXP a, SEXP b, ...)
```

R and C/C++

A good fit, it turns out (cont.)

- An **SEXP** (or S-Expression Pointer) is used for *everything*
- (An older C trick approximating object-oriented programming)
- We can ignore the details but retain that
    - everything in R is a **SEXP**, and the **SEXP** is self-describing
    - can matrix, vector, list, function, ... for a total of 27 types
- The key thing for Rcpp is that via C++ features we can map
    - each of these (few) **SEXP** types to a specific C++ class representing that type
    - and the conversion is automated back and forth

R and C/C++

Other good reasons

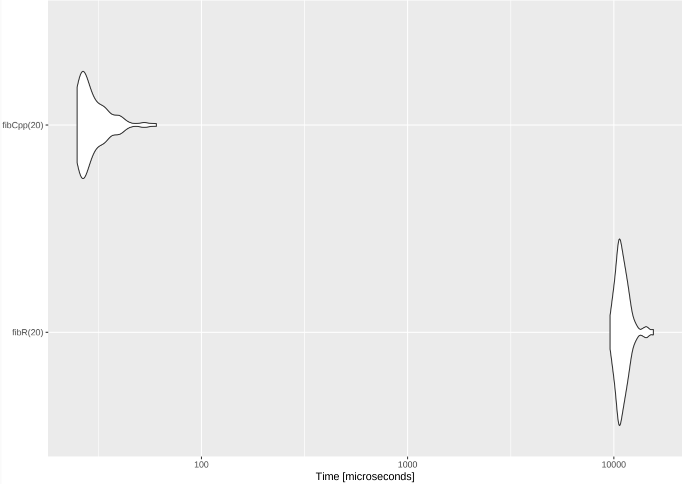
- It is *fast* – compiled C++ is hard to beat in other languages
  - (That said, you can *of course* write bad and slow code....)
- It is *very general* and widely used
  - *many libraries*
  - many tools
- It is fairly universal:
  - just about anything will have C interface so C++ can play
  - just about any platform / OS will have it

R and C/C++

Key Features

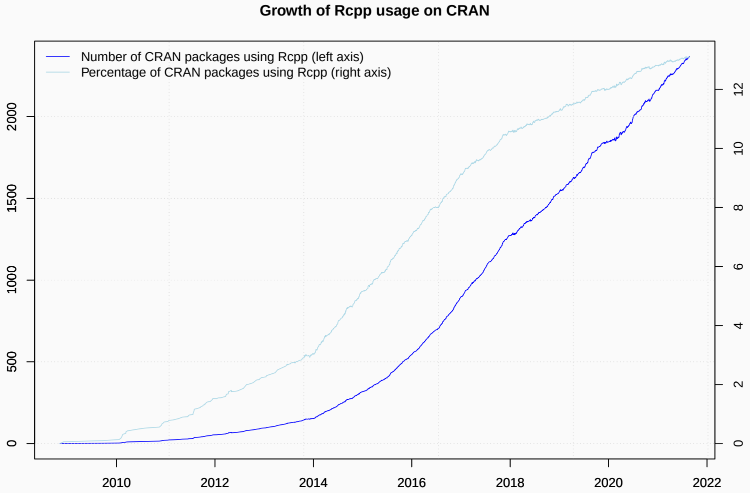
- (Fairly) Easy to learn as it really does not have to be that complicated there are numerous examples, tutorials, books, ...
- Easy to use as it avoids build and operating system build complexities simply by relying the robust build infrastrucure R itself has
- Expressive as it allows for *vectorised* C++ using *Rcpp Sugar*
- Seamless access to all R objects: vector, matrix, list, S3/S4/RefClass, Function, ...
- Fast as Rcpp excels at tasks where R struggles: loops, function calls, ...
- Extensions facilitates access to external libraries directly or via eg *Rcpp modules*

Rcpp Speed Illustration



Benchmark on Fibonacci(20) between C++ and R – note the log scale!

Growth



Source: Our calculations. Data current as of August 29, 2021.

Users on Core Repositories

Rcpp is currently used by

- 2367 CRAN packages
- 231 BioConductor packages
- an unknown (but “large”) number of GitHub projects
- over 50 million downloads (from the RStudio operated CRAN mirrors alone)

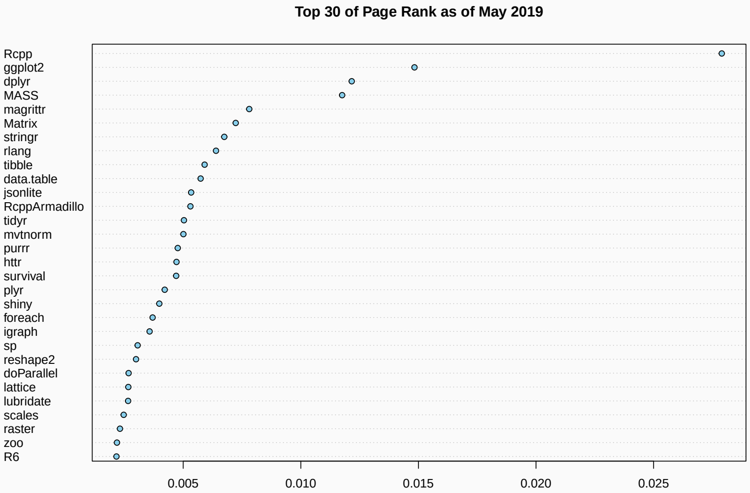
Pagerank

```
suppressMessages(library(utils))
library(pagerank)   # cf github.com/andrie/pagerank

cran <- ”http://cloud.r-project.org”
pr <- compute_pagerank(cran)
round(100*pr[1:5], 3)
```

```
##     Rcpp  ggplot2    dplyr     MASS magrittr
##    2.790    1.483    1.216    1.176    0.781
```

PageRank



Top 30 of Page Rank as of May 2019
Rcpp
ggplot2
dplyr
MASS
magrittr
Matrix
stringr
rlang
tibble
data.table
jsonlite
RcppArmadillo
tidyr
mvtnorm
purrr
httr
survival
plyr
shiny
foreach
igraph
sp
reshape2
doParallel
lattice
lubridate
scales
raster
zoo
R6
0.005
0.010
0.015
0.020
0.025

Percentage of Compiled Packages

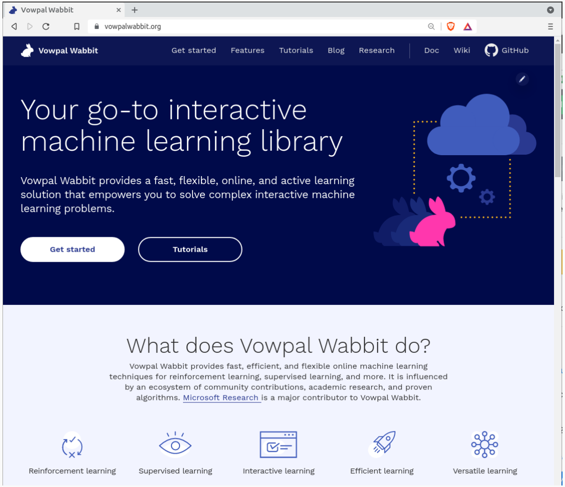
```r
db <- tools::CRAN_package_db()
nTot <- nrow(db)
## all direct Rcpp reverse depends, ie packages using Rcpp
nRcpp <- length(tools::dependsOnPkgs(”Rcpp”, recursive=FALSE,installed=db))
nCompiled <- table(db[, ”NeedsCompilation”])[[”yes”]]
propTot <- nRcpp / nTot * 100
propComp <- nRcpp / nCompiled * 100
data.frame(tot=nTot, totRcpp = nRcpp, totCompiled = nCompiled,
           RcppPctTot = propTot, RcppPctOfCompiled = propComp)
```

```
##     tot totRcpp totCompiled RcppPctTot RcppPctOfCompiled
## 1 18112    2367        4338    13.0687           54.5643
```

Machine Learning via R(cpp): Three Examples

Vowpal Wabbit

Vowpal Wabbit



Your go-to interactive machine learning library

Vowpal Wabbit provides a fast, flexible, online, and active learning solution that empowers you to solve complex interactive machine learning problems.

What does Vowpal Wabbit do?

Vowpal Wabbit provides fast, efficient, and flexible online machine learning techniques for reinforcement learning, supervised learning, and more. It is influenced by an ecosystem of community contributions, academic research, and proven algorithms. Microsoft Research is a major contributor to Vowpal Wabbit.

Well-known, established project

Initially at Yahoo! Research, now at Microsoft Research (same NY lab)

Lead by John Langford

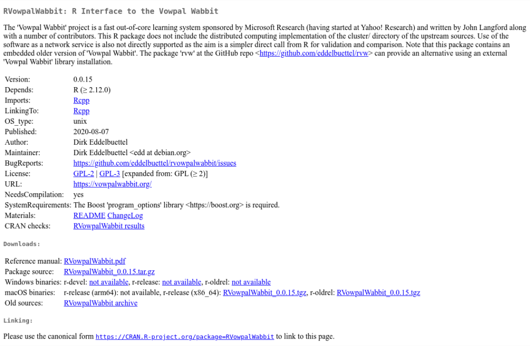
Fast online learner, popular as a classifier

Now much more included distributed setups

Many papers, tutorials, ...

C++ core

Vowpal Wabbit in R



RVowpalWabbit: R Interface to the Vowpal Wabbit

The 'Vowpal Wabbit' project is a fast out-of-core learning system sponsored by Microsoft Research (having started at Yahoo! Research) and written by John Langford along with a number of contributors. This R package does not include the distributed computing implementation of the cluster/ directory of the upstream sources. Use of the software as a network service is also not directly supported as the aim is a simpler direct call from R for validation and comparison. Note that this package contains an embedded older version of 'Vowpal Wabbit'. The package 'rvw' at the GitHub repo <https://github.com/eddelbuettel/rvw> can provide an alternative using an external 'Vowpal Wabbit' library installation.

Version:	0.0.15
Depends:	R (≥ 2.12.0)
Imports:	Rcpp
LinkingTo:	Rcpp
OS_type:	unix
Published:	2020-08-07
Author:	Dirk Eddelbuettel
Maintainer:	Dirk Eddelbuettel <edd at debian.org>
BugReports:	https://github.com/eddelbuettel/rvowpalwabbit/issues
License:	GPL-2 | GPL-3 [expanded from: GPL (≥ 2)]
URL:	https://vowpalwabbit.org/
NeedsCompilation:	yes
SystemRequirements:	The Boost 'program_options' library <https://boost.org> is required.
Materials:	README ChangeLog
CRAN checks:	RVowpalWabbit results

Downloads:

Reference manual:	RVowpalWabbit.pdf
Package source:	RVowpalWabbit_0.0.15.tar.gz
Windows binaries:	r-devel: not available, r-release: not available, r-oldrel: not available
macOS binaries:	r-release (arm64): not available, r-release (x86_64): RVowpalWabbit_0.0.15.tgz, r-oldrel: RVowpalWabbit_0.0.15.tgz
Old sources:	RVowpalWabbit archive

Linking:

Please use the canonical form `https://CRAN.R-project.org/package=RVowpalWabbit` to link to this page.

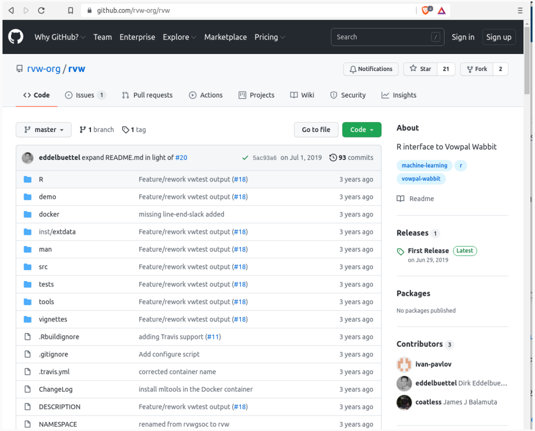
Several Phases

Initial package **`RVowpalWabbit`** on CRAN since Sep 2011 (!!), took a copy of the library (plus some editing)

At the time 'VW' library had an awkward / unusual build

Taking a copy is quick and simple … and can leads to stale code

Vowpal Wabbit in R



rvw-org / rvw

eddelbuettel expand README.md in light of #20 ✓ 5ac93a6 on Jul 1, 2019 93 commits

R	Feature/rework vwtest output (#18)	3 years ago
demo	Feature/rework vwtest output (#18)	3 years ago
docker	missing line-end-slack added	3 years ago
inst/extdata	Feature/rework vwtest output (#18)	3 years ago
man	Feature/rework vwtest output (#18)	3 years ago
src	Feature/rework vwtest output (#18)	3 years ago
tests	Feature/rework vwtest output (#18)	3 years ago
tools	Feature/rework vwtest output (#18)	3 years ago
vignettes	Feature/rework vwtest output (#18)	3 years ago
.Rbuildignore	adding Travis support (#11)	3 years ago
.gitignore	Add configure script	3 years ago
.travis.yml	corrected container name	3 years ago
ChangeLog	install mltools in the Docker container	3 years ago
DESCRIPTION	Feature/rework vwtest output (#18)	3 years ago
NAMESPACE	renamed from rvwgsoc to rvw	3 years ago

About

R interface to Vowpal Wabbit

machine-learning r vowpal-wabbit

Readme

Releases 1

First Release Latest on Jun 29, 2019

Packages

No packages published

Contributors 3

ivan-pavlov

eddelbuettel Dirk Eddelbue…

coatless James J Balamuta

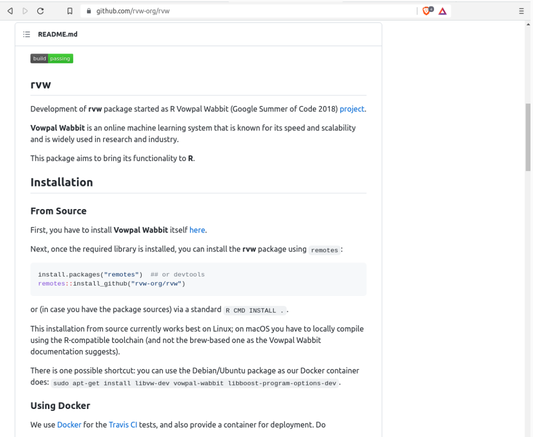
Second Version

Google Summer of Code project by Ivan Pavlov (supervised by DE + JJB) building on earlier work by Selim Raboudi

Used external 'VW' library (which now build better)

Taking an external library is more elegant … but harder on the user who needs to install it – so for example *not* at CRAN

Vowpal Wabbit in R



rvw

Development of **rvw** package started as R Vowpal Wabbit (Google Summer of Code 2018) project.

Vowpal Wabbit is an online machine learning system that is known for its speed and scalability and is widely used in research and industry.

This package aims to bring its functionality to **R**.

Installation

From Source

First, you have to install **Vowpal Wabbit** itself here.

Next, once the required library is installed, you can install the **rvw** package using `remotes`:

```
install.packages("remotes")  ## or devtools
remotes::install_github("rvw-org/rvw")
```

or (in case you have the package sources) via a standard `R CMD INSTALL .`.

This installation from source currently works best on Linux; on macOS you have to locally compile using the R-compatible toolchain (and not the brew-based one as the Vowpal Wabbit documentation suggests).

There is one possible shortcut: you can use the Debian/Ubuntu package as our Docker container does: `sudo apt-get install libvw-dev vowpal-wabbit libboost-program-options-dev`.

Using Docker

We use Docker for the Travis CI tests, and also provide a container for deployment. Do

Second Version

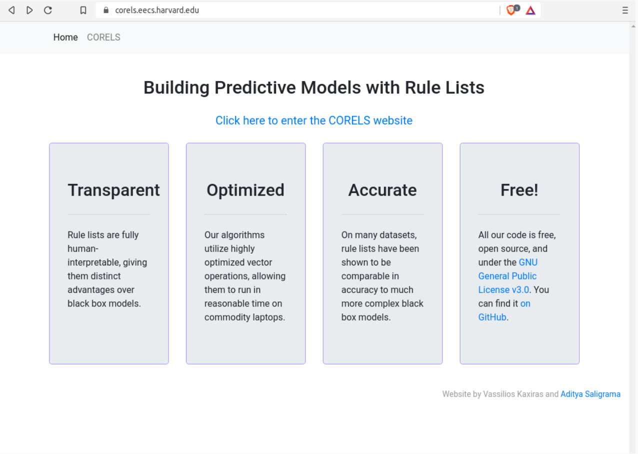
It is however in a pretty good state

Contributions would be welcome, a project worth revisiting

As a fallback and alternative build approach, we could internalise building the library in order to get the package onto CRAN

Corels

Corels



Building Predictive Models with Rule Lists

Click here to enter the CORELS website

Transparent

Rule lists are fully human-interpretable, giving them distinct advantages over black box models.

Optimized

Our algorithms utilize highly optimized vector operations, allowing them to run in reasonable time on commodity laptops.

Accurate

On many datasets, rule lists have been shown to be comparable in accuracy to much more complex black box models.

Free!

All our code is free, open source, and under the GNU General Public License v3.0. You can find it on GitHub.

Website by Vassilios Kaxiras and Aditya Saligrama

Very nice (smaller) project out of Harvard/Duke/UBC

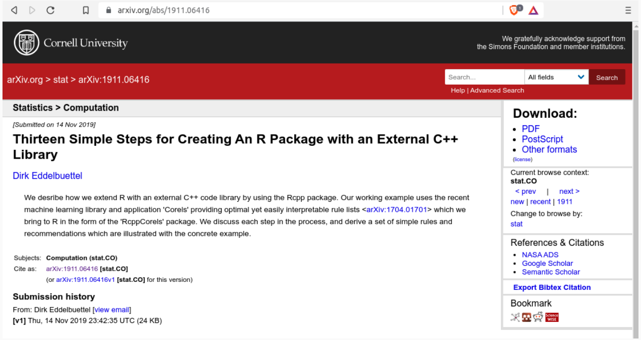
Well documented with a couple of papers

Lead by Margo Seltzer and Cynthia Rudin

I wrote a simple wrapper package, and *documented my steps* in a what became a small paper

C++ core

Corels in R



arxiv.org/abs/1911.06416

Cornell University

We gratefully acknowledge support from
the Simons Foundation and member institutions.

arXiv.org > stat > arXiv:1911.06416

Search... | All fields | Search

Help | Advanced Search

Statistics > Computation

[Submitted on 14 Nov 2019]

Thirteen Simple Steps for Creating An R Package with an External C++ Library

Dirk Eddelbuettel

We desribe how we extend R with an external C++ code library by using the Rcpp package. Our working example uses the recent machine learning library and application 'Corels' providing optimal yet easily interpretable rule lists <arXiv:1704.01701> which we bring to R in the form of the 'RcppCorels' package. We discuss each step in the process, and derive a set of simple rules and recommendations which are illustrated with the concrete example.

Subjects: **Computation (stat.CO)**
Cite as: arXiv:1911.06416 **[stat.CO]**
(or arXiv:1911.06416v1 **[stat.CO]** for this version)

Submission history

From: Dirk Eddelbuettel [view email]
[v1] Thu, 14 Nov 2019 23:42:35 UTC (24 KB)

Download:

- PDF
- PostScript
- Other formats

(license)

Current browse context:
stat.CO
< prev | next >
new | recent | 1911
Change to browse by:
stat

References & Citations

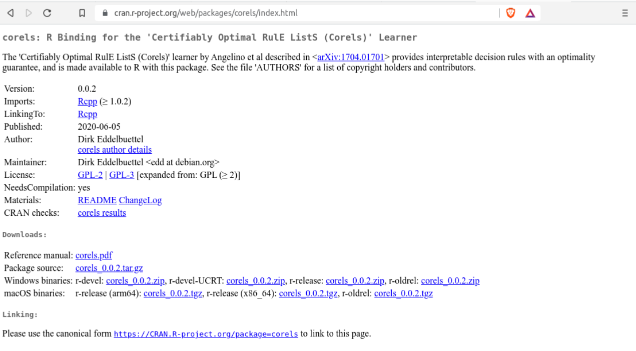
- NASA ADS
- Google Scholar
- Semantic Scholar

Export Bibtex Citation

Bookmark

- Little how-to tutorial as a side effect of wrapping Corels into RcppCorels
- Paper now also a vignette in the Rcpp package
- RcppCorels intgegrated with the rest of Corels at `https://github.com/corels`

Corels in R



corels: R Binding for the 'Certifiably Optimal RulE ListS (Corels)' Learner

The 'Certifiably Optimal RulE ListS (Corels)' learner by Angelino et al described in <arXiv:1704.01701> provides interpretable decision rules with an optimality guarantee, and is made available to R with this package. See the file 'AUTHORS' for a list of copyright holders and contributors.

Version:	0.0.2
Imports:	Rcpp (≥ 1.0.2)
LinkingTo:	Rcpp
Published:	2020-06-05
Author:	Dirk Eddelbuettel corels author details
Maintainer:	Dirk Eddelbuettel <edd at debian.org>
License:	GPL-2 | GPL-3 [expanded from: GPL (≥ 2)]
NeedsCompilation:	yes
Materials:	README ChangeLog
CRAN checks:	corels results

Downloads:

Reference manual:	corels.pdf
Package source:	corels_0.0.2.tar.gz
Windows binaries:	r-devel: corels_0.0.2.zip, r-devel-UCRT: corels_0.0.2.zip, r-release: corels_0.0.2.zip, r-oldrel: corels_0.0.2.zip
macOS binaries:	r-release (arm64): corels_0.0.2.tgz, r-release (x86_64): corels_0.0.2.tgz, r-oldrel: corels_0.0.2.tgz

Linking:

Please use the canonical form `https://CRAN.R-project.org/package=corels` to link to this page.

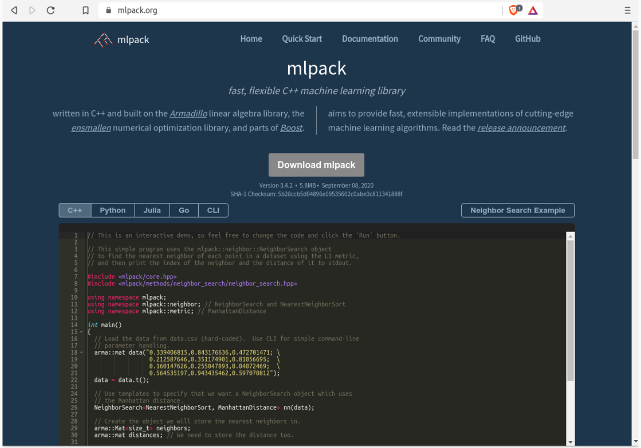
Now also on CRAN as `https://cran.r-project.org/package=corels`

(with the package name normalised to `corels`)

Some work remaining to integrate better with external data sets

MLPACK

MLPACK



mlpack

fast, flexible C++ machine learning library

written in C++ and built on the *Armadillo* linear algebra library, the *ensmallen* numerical optimization library, and parts of *Boost*.

aims to provide fast, extensible implementations of cutting-edge machine learning algorithms. Read the *release announcement*.

Version 3.4.2 • 5.8MB • September 08, 2020
SHA-1 Checksum: 5b28ccb5d04896e09535602c0abe0c811341888f

```cpp
// This is an interactive demo, so feel free to change the code and click the 'Run' button.

// This simple program uses the mlpack::neighbor::NeighborSearch object
// to find the nearest neighbor of each point in a dataset using the L1 metric,
// and then print the index of the neighbor and the distance of it to stdout.

#include <mlpack/core.hpp>
#include <mlpack/methods/neighbor_search/neighbor_search.hpp>

using namespace mlpack;
using namespace mlpack::neighbor; // NeighborSearch and NearestNeighborSort
using namespace mlpack::metric; // ManhattanDistance

int main()
{
  // Load the data from data.csv (hard-coded).  Use CLI for simple command-line
  // parameter handling.
  arma::mat data("0.339406815,0.843176636,0.472701471; \
                  0.212587646,0.351174901,0.81056695;  \
                  0.160147626,0.255047893,0.04072469;  \
                  0.564535197,0.943435462,0.597070812");
  data = data.t();

  // Use templates to specify that we want a NeighborSearch object which uses
  // the Manhattan distance.
  NeighborSearch<NearestNeighborSort, ManhattanDistance> nn(data);

  // Create the object we will store the nearest neighbors in.
  arma::Mat<size_t> neighbors;
  arma::mat distances; // We need to store the distance too.

```

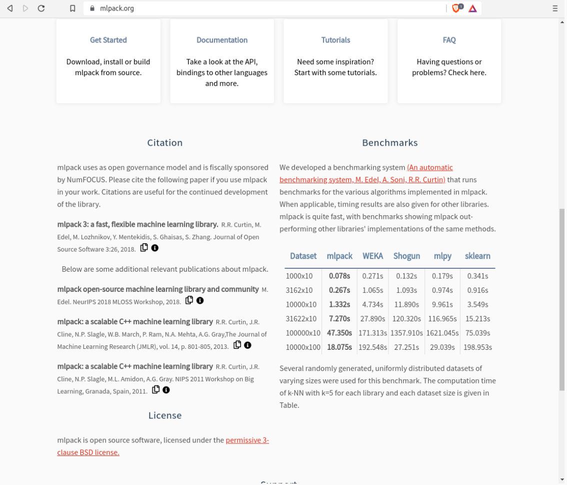
Mature and established project

Lead by Ryan Curtin with many contributors

Very complete collection of algorithm

Multiple language bindings

MLPACK



Get Started

Download, install or build mlpack from source.

Documentation

Take a look at the API, bindings to other languages and more.

Tutorials

Need some inspiration? Start with some tutorials.

FAQ

Having questions or problems? Check here.

Citation

mlpack uses as open governance model and is fiscally sponsored by NumFOCUS. Please cite the following paper if you use mlpack in your work. Citations are useful for the continued development of the library.

mlpack 3: a fast, flexible machine learning library. R.R. Curtin, M. Edel, M. Lozhnikov, Y. Mentekidis, S. Ghaisas, S. Zhang. Journal of Open Source Software 3:26, 2018.

Below are some additional relevant publications about mlpack.

mlpack open-source machine learning library and community M. Edel. NeurIPS 2018 MLOSS Workshop, 2018.

mlpack: a scalable C++ machine learning library R.R. Curtin, J.R. Cline, N.P. Slagle, W.B. March, P. Ram, N.A. Mehta, A.G. Gray,The Journal of Machine Learning Research (JMLR), vol. 14, p. 801-805, 2013.

mlpack: a scalable C++ machine learning library R.R. Curtin, J.R. Cline, N.P. Slagle, M.L. Amidon, A.G. Gray. NIPS 2011 Workshop on Big Learning, Granada, Spain, 2011.

License

mlpack is open source software, licensed under the permissive 3-clause BSD license.

Benchmarks

We developed a benchmarking system (An automatic benchmarking system, M. Edel, A. Soni, R.R. Curtin) that runs benchmarks for the various algorithms implemented in mlpack. When applicable, timing results are also given for other libraries. mlpack is quite fast, with benchmarks showing mlpack out-performing other libraries' implementations of the same methods.

Dataset	mlpack	WEKA	Shogun	mlpy	sklearn
1000x10	**0.078s**	0.271s	0.132s	0.179s	0.341s
3162x10	**0.267s**	1.065s	1.093s	0.974s	0.916s
10000x10	**1.332s**	4.734s	11.890s	9.961s	3.549s
31622x10	**7.270s**	27.890s	120.320s	116.965s	15.213s
100000x10	**47.350s**	171.313s	1357.910s	1621.045s	75.039s
10000x100	**18.075s**	192.548s	27.251s	29.039s	198.953s

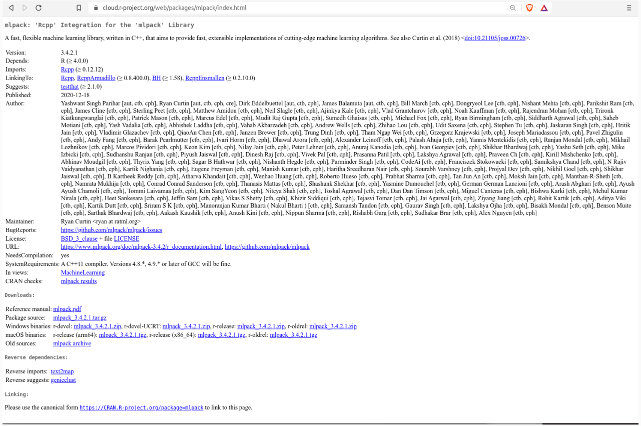
Several randomly generated, uniformly distributed datasets of varying sizes were used for this benchmark. The computation time of k-NN with k=5 for each library and each dataset size is given in Table.

Many models implemented

(Run-time) Performance usually better than competitors in Java or Python

C++ core

MLPACK IN R



mlpack: 'Rcpp' Integration for the 'mlpack' Library

A fast, flexible machine learning library, written in C++, that aims to provide fast, extensible implementations of cutting-edge machine learning algorithms. See also Curtin et al. (2018) <doi:10.21105/joss.00726>.

Version:	3.4.2.1
Depends:	R (≥ 4.0.0)
Imports:	Rcpp (≥ 0.12.12)
LinkingTo:	Rcpp, RcppArmadillo (≥ 0.8.400.0), BH (≥ 1.58), RcppEnsmallen (≥ 0.2.10.0)
Suggests:	testthat (≥ 2.1.0)
Published:	2020-12-18
Author:	Yashwant Singh Parihar [aut, ctb, cph], Ryan Curtin [aut, ctb, cph, cre], Dirk Eddelbuettel [aut, ctb, cph], James Balamuta [aut, ctb, cph], Bill March [ctb, cph], Dongryeol Lee [ctb, cph], Nishant Mehta [ctb, cph], Parikshit Ram [ctb, cph], James Cline [ctb, cph], Sterling Peet [ctb, cph], Matthew Amidon [ctb, cph], Neil Slagle [ctb, cph], Ajinkya Kale [ctb, cph], Vlad Grantcharov [ctb, cph], Noah Kauffman [ctb, cph], Rajendran Mohan [ctb, cph], Trironk Kiatkungwanglai [ctb, cph], Patrick Mason [ctb, cph], Marcus Edel [ctb, cph], Mudit Raj Gupta [ctb, cph], Sumedh Ghaisas [ctb, cph], Michael Fox [ctb, cph], Ryan Birmingham [ctb, cph], Siddharth Agrawal [ctb, cph], Saheb Motiani [ctb, cph], Yash Vadalia [ctb, cph], Abhishek Laddha [ctb, cph], Vahab Akbarzadeh [ctb, cph], Andrew Wells [ctb, cph], Zhihao Lou [ctb, cph], Udit Saxena [ctb, cph], Stephen Tu [ctb, cph], Jaskaran Singh [ctb, cph], Hritik Jain [ctb, cph], Vladimir Glazachev [ctb, cph], QiaoAn Chen [ctb, cph], Janzen Brewer [ctb, cph], Trung Dinh [ctb, cph], Tham Ngap Wei [ctb, cph], Grzegorz Krajewski [ctb, cph], Joseph Mariadassou [ctb, cph], Pavel Zhigulin [ctb, cph], Andy Fang [ctb, cph], Barak Pearlmutter [ctb, cph], Ivari Horm [ctb, cph], Dhawal Arora [ctb, cph], Alexander Leinoff [ctb, cph], Palash Ahuja [ctb, cph], Yannis Mentekidis [ctb, cph], Ranjan Mondal [ctb, cph], Mikhail Lozhnikov [ctb, cph], Marcos Pividori [ctb, cph], Keon Kim [ctb, cph], Nilay Jain [ctb, cph], Peter Lehner [ctb, cph], Anuraj Kanodia [ctb, cph], Ivan Georgiev [ctb, cph], Shikhar Bhardwaj [ctb, cph], Yashu Seth [ctb, cph], Mike Izbicki [ctb, cph], Sudhanshu Ranjan [ctb, cph], Piyush Jaiswal [ctb, cph], Dinesh Raj [ctb, cph], Vivek Pal [ctb, cph], Prasanna Patil [ctb, cph], Lakshya Agrawal [ctb, cph], Praveen Ch [ctb, cph], Kirill Mishchenko [ctb, cph], Abhinav Moudgil [ctb, cph], Thyrix Yang [ctb, cph], Sagar B Hathwar [ctb, cph], Nishanth Hegde [ctb, cph], Parminder Singh [ctb, cph], CodeAi [ctb, cph], Franciszek Stokowacki [ctb, cph], Samikshya Chand [ctb, cph], N Rajiv Vaidyanathan [ctb, cph], Kartik Nighania [ctb, cph], Eugene Freyman [ctb, cph], Manish Kumar [ctb, cph], Haritha Sreedharan Nair [ctb, cph], Sourabh Varshney [ctb, cph], Projyal Dev [ctb, cph], Nikhil Goel [ctb, cph], Shikhar Jaiswal [ctb, cph], B Kartheek Reddy [ctb, cph], Atharva Khandait [ctb, cph], Wenhao Huang [ctb, cph], Roberto Hueso [ctb, cph], Prabhat Sharma [ctb, cph], Tan Jun An [ctb, cph], Moksh Jain [ctb, cph], Manthan-R-Sheth [ctb, cph], Namrata Mukhija [ctb, cph], Conrad Conrad Sanderson [ctb, cph], Thanasis Mattas [ctb, cph], Shashank Shekhar [ctb, cph], Yasmine Dumouchel [ctb, cph], German German Lancioni [ctb, cph], Arash Abghari [ctb, cph], Ayush Chamoli [ctb, cph], Tommi Laivamaa [ctb, cph], Kim SangYeon [ctb, cph], Niteya Shah [ctb, cph], Toshal Agrawal [ctb, cph], Dan Dan Timson [ctb, cph], Miguel Canteras [ctb, cph], Bishwa Karki [ctb, cph], Mehul Kumar Nirala [ctb, cph], Heet Sankesara [ctb, cph], Jeffin Sam [ctb, cph], Vikas S Shetty [ctb, cph], Khizir Siddiqui [ctb, cph], Tejasvi Tomar [ctb, cph], Jai Agarwal [ctb, cph], Ziyang Jiang [ctb, cph], Rohit Kartik [ctb, cph], Aditya Viki [ctb, cph], Kartik Dutt [ctb, cph], Sriram S K [ctb, cph], Manoranjan Kumar Bharti (Nakul Bharti) [ctb, cph], Saraansh Tandon [ctb, cph], Gaurav Singh [ctb, cph], Lakshya Ojha [ctb, cph], Bisakh Mondal [ctb, cph], Benson Muite [ctb, cph], Sarthak Bhardwaj [ctb, cph], Aakash Kaushik [ctb, cph], Anush Kini [ctb, cph], Nippun Sharma [ctb, cph], Rishabh Garg [ctb, cph], Sudhakar Brar [ctb, cph], Alex Nguyen [ctb, cph]
Maintainer:	Ryan Curtin <ryan at ratml.org>
BugReports:	https://github.com/mlpack/mlpack/issues
License:	BSD_3_clause + file LICENSE
URL:	https://www.mlpack.org/doc/mlpack-3.4.2/r_documentation.html, https://github.com/mlpack/mlpack
NeedsCompilation:	yes
SystemRequirements:	A C++11 compiler. Versions 4.8.*, 4.9.* or later of GCC will be fine.
In views:	MachineLearning
CRAN checks:	mlpack results

Downloads:

Reference manual:	mlpack.pdf
Package source:	mlpack_3.4.2.1.tar.gz
Windows binaries:	r-devel: mlpack_3.4.2.1.zip, r-devel-UCRT: mlpack_3.4.2.1.zip, r-release: mlpack_3.4.2.1.zip, r-oldrel: mlpack_3.4.2.1.zip
macOS binaries:	r-release (arm64): mlpack_3.4.2.1.tgz, r-release (x86_64): mlpack_3.4.2.1.tgz, r-oldrel: mlpack_3.4.2.1.tgz
Old sources:	mlpack archive

Reverse dependencies:

Reverse imports:	text2map
Reverse suggests:	genieclust

Linking:

Please use the canonical form `https://CRAN.R-project.org/package=mlpack` to link to this page.

Several earlier attempts at creating a package stopped for lack of a (external) library (say at CRAN)

Excellent Google Summer of Code work by Yashwant Singh Parihar in 2020; R is now a first-class supported language

Also on CRAN as `https://cran.r-project.org/package=mlpack`

Summary

R for Machine Learning

Key Points

- R excels as *data-centric language* invented by statisticians
- Interfaces to other systems are key part of its design
- Many promising libraries either are in C++, or have C++ interface
- The Rcpp smoothes the interaction with another (compiled) language
- We looked at three concrete examples:
  - Vowpal Wabbit
  - Corels
  - MLPACK

Thank you!

slides	https://dirk.eddelbuettel.com/presentations/
web	https://dirk.eddelbuettel.com/
mail	dirk@eddelbuettel.com
github	@eddelbuettel
twitter	@eddelbuettel