Package: RcppGSL (via r-universe)

October 28, 2026

Type Package

Title 'Rcpp' Integration for 'GNU GSL' Vectors and Matrices

Version 0.3.14

Date 2026-03-05

Description 'Rcpp' integration for 'GNU GSL' vectors and matrices The 'GNU Scientific Library' (or 'GSL') is a collection of numerical routines for scientific computing. It is particularly useful for C and C++ programs as it provides a standard C interface to a wide range of mathematical routines. There are over 1000 functions in total with an extensive test suite. The 'RcppGSL' package provides an easy-to-use interface between 'GSL' data structures and R using concepts from 'Rcpp' which is itself a package that eases the interfaces between R and C++. This package also serves as a prime example of how to build a package that uses 'Rcpp' to connect to another third-party library. The 'autoconf' script, 'inline' plugin and example package can all be used as a stanza to write a similar package against another library.

License GPL (>= 2)

LazyLoad yes

LinkingTo Rcpp

Imports Rcpp (>= 1.1.1), stats

Suggests tinytest

SystemRequirements GNU GSL

URL https://github.com/eddelbuettel/rcppgsl, https://dirk.eddelbuettel.com/code/rcpp.gsl.html

BugReports https://github.com/eddelbuettel/rcppgsl/issues

Encoding UTF-8

VignetteBuilder Rcpp

NeedsCompilation yes

Author Dirk Eddelbuettel [aut, cre] (ORCID:
<https://orcid.org/0000-0001-6419-907X>), Romain Francois [aut]
(ORCID: <https://orcid.org/0000-0002-2444-4226>)

Maintainer Dirk Eddelbuettel <edd@debian.org>

Config/pak/sysreqs libgsl0-dev

Repository https://cran.r-universe.dev

Date/Publication 2026-03-05 15:20:08 UTC

RemoteUrl https://github.com/cran/RcppGSL

RemoteRef HEAD

RemoteSha a2514f20b7d49ceaaa22059ce0796c9e139d5261

Contents

RcppGSL-package 2
fastLm 3
LdFlags 4

Index **6**

`RcppGSL-package` *'Rcpp' Integration for 'GNU GSL' Vectors and Matrices*

Description

'Rcpp' integration for 'GNU GSL' vectors and matrices The 'GNU Scientific Library' (or 'GSL') is a collection of numerical routines for scientific computing. It is particularly useful for C and C++ programs as it provides a standard C interface to a wide range of mathematical routines. There are over 1000 functions in total with an extensive test suite. The 'RcppGSL' package provides an easy-to-use interface between 'GSL' data structures and R using concepts from 'Rcpp' which is itself a package that eases the interfaces between R and C++. This package also serves as a prime example of how to build a package that uses 'Rcpp' to connect to another third-party library. The 'autoconf' script, 'inline' plugin and example package can all be used as a stanza to write a similar package against another library.

Maintainer

Dirk Eddelbuettel <edd@debian.org>

Author(s)

Dirk Eddelbuettel [aut, cre] (ORCID: <https://orcid.org/0000-0001-6419-907X>), Romain Francois [aut] (ORCID: <https://orcid.org/0000-0002-2444-4226>)

References

GSL: GNU Scientific Library: http://www.gnu.org/software/gsl/

fastLm *Bare-bones linear model fitting function*

Description

fastLm estimates the linear model using the gsl_multifit_linear function of the GNU GSL library.

Usage

```
fastLmPure(X, y)

fastLm(X, ...)
## Default S3 method:
fastLm(X, y, ...)
## S3 method for class 'formula'
fastLm(formula, data = list(), ...)
```

Arguments

y	a vector containing the explained variable.
X	a model matrix.
formula	a symbolic description of the model to be fit.
data	an optional data frame containing the variables in the model.
...	not used

Details

Linear models should be estimated using the lm function. In some cases, lm.fit may be appropriate.

The fastLmPure function provides a reference use case of the GSL library via the wrapper functions in the **RcppGSL** package.

The fastLm function provides a more standard implementation of a linear model fit, offering both a default and a formula interface as well as print, summary and predict methods.

Lastly, one must be be careful in timing comparisons of lm and friends versus this approach based on GSL or Armadillo. The reason that GSL or Armadillo can do something like lm.fit faster than the functions in the stats package is because they use the Lapack version of the QR decomposition while the stats package uses a *modified* Linpack version. Hence GSL and Armadillo uses level-3 BLAS code whereas the stats package uses level-1 BLAS. However, GSL or Armadillo will choke on rank-deficient model matrices whereas the functions from the stats package will handle them properly due to the modified Linpack code. Statisticians want a pivoting scheme of "pivot only on (apparent) rank deficiency" and numerical analysts have no idea why statisticians want this so it is not part of conventional linear algebra software.

Value

`fastLmPure` returns a list with three components:

`coefficients`	a vector of coefficients
`stderr`	a vector of the (estimated) standard errors of the coefficient estimates
`df`	a scalar denoting the degrees of freedom in the model

`fastLm` returns a richer object which also includes the residuals and call similar to the `lm` or `rlm` functions..

Author(s)

The GNU GSL library is being written by team of authors with the overall development, design and implementation lead by Brian Gough and Gerard Jungman. RcppGSL is written by Romain Francois and Dirk Eddelbuettel.

References

GNU GSL project: https://www.gnu.org/software/gsl/

See Also

`lm`, `lm.fit`

Examples

```
data(trees, package="datasets")

## bare-bones direct interface
flm <- fastLmPure( cbind(1, log(trees$Girth)), log(trees$Volume) )
print(flm)

## standard R interface for formula or data returning object of class fastLm
flmmod <- fastLm( log(Volume) ~ log(Girth), data=trees)
summary(flmmod)
```

LdFlags *Provide RcppGSL Compiler and Linker Flags*

Description

`LdFlags` and `CFlags` return the required flags and options for the compiler and system linker in order to build against GNU GSL. This allows portable use of **RcppGSL** (which needs the GNU GSL) as package location as well as operating-system specific details are abstracted away behind the interface of this function.

`LdFlags` and `CFlags` are commonly called from the files `Makevars` (or `Makevars.win`) rather than in an interactive session.

Usage

```
LdFlags(print=TRUE)
CFlags(print=TRUE)
```

Arguments

print	A boolean determining whether the requested value is returned on the standard output, or silenly as a value.

Details

Thee functions are not meant to used interactively, and are intended solely for use by the build tools.

The values that are returned are acquired by the package at load time. On Linux and OS X, the `pkg-config` program is queried. On Windows, environment variables used for GNU GSL builds with R are used.

Value

A character vector suitable by use by the system compiler linker in order to compile and/or link against the GNU GSK.

Author(s)

Dirk Eddelbuettel and Romain Francois

References

Dirk Eddelbuettel and Romain Francois (2011). **Rcpp**: Seamless R and C++ Integration. *Journal of Statistical Software*, **40(8)**, 1-18. URL http://www.jstatsoft.org/v40/i08/ and available as `vignette("Rcpp-introduction")`.

See Also

The document of the `pkg-config` system tool.

Index

∗ **interface**
LdFlags, 4
∗ **package**
RcppGSL-package, 2
∗ **programming**
LdFlags, 4
∗ **regression**
fastLm, 3

CFlags *(*LdFlags*)*, 4

fastLm, 3
fastLmPure *(*fastLm*)*, 3

LdFlags, 4
lm, *3, 4*
lm.fit, *3, 4*

RcppGSL *(*RcppGSL-package*)*, 2
RcppGSL-package, 2
rlm, *4*