

Package: lsReg (via r-universe)

October 3, 2026

Type Package

Title Performs Large Scale Regressions

Version 1.0.0

Date 2026-04-07

Maintainer John Morrison <jmorr@usc.edu>

Description Routines to perform large scale regression. Linear, logistic, and Poisson regressions are supported. Large scale regression efficiently fits models where a small number of covariates are changing and the subjects have complete data. A genome wide association study would be an example.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

Suggests testthat (>= 3.0.0), knitr, rmarkdown, statmod

VignetteBuilder knitr

RoxygenNote 7.3.3

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp, methods

Config/testthat/edition 3

NeedsCompilation yes

Author John Morrison [aut, cre], NCI [fnd] (CA196559), NCI [fnd] (CA201407), NIEHS [fnd] (ES007048), NHLBI [fnd] (HL115606)

Repository <https://jimb3.r-universe.dev>

Date/Publication 2026-04-28 20:45:30 UTC

RemoteUrl <https://github.com/cran/lsReg>

RemoteRef HEAD

RemoteSha f15a96ec39f3cbb324bab581a75ec1409246cd9c

Contents

addcovar	2
lsReg	3
Index	4

addcovar	<i>Run a large-scale regression test</i>
----------	--

Description

Computes a hypothesis test statistic for one or more new covariates `xr` using memory pre-allocated by `lsReg`.

Usage

```
addcovar(lsregmem, xr)
```

Arguments

`lsregmem` An object of class `lsregmem` as returned by `lsReg`.

`xr` Numeric matrix of additional covariates to test. Number of columns must match the `colstoadd` value used in `lsReg`.

Value

Invisibly returns the exit code (0 on success, nonzero on error). After a successful call, results are stored in the `lsregmem` object:

`lsregmem$testvalue` The test statistic. For "lrt" this is a chi-square statistic (p-values via `pchisq`). For all other test types this is a z-score (p-values via `pnorm`).

`lsregmem$fitdata$betab` The parameter estimate(s) for `xr`. Not meaningful for "score" or "robustscore", which do not fit the full model.

Examples

```
datafile <- system.file("extdata", "simulated_data.rds", package = "lsReg")
dat <- readRDS(datafile)
basemdl <- glm(ylin ~ x1 + x2, data = dat, family = gaussian)
mem <- lsReg(basemdl, colstoadd = 1, testtype = "wald")
addcovar(mem, as.matrix(dat[, "z5", drop = FALSE]))
mem$fitdata$betab[1] # parameter estimate for z5
mem$testvalue[1, 1] # Wald z-score for z5
```

`lsReg`*Allocate memory for large-scale regression*

Description

Prepares and caches data structures from a fitted base GLM for use in repeated large-scale hypothesis tests via [addcovar](#).

Usage

```
lsReg(basemdl, colstoadd, testtype)
```

Arguments

<code>basemdl</code>	Base model of the form $y \sim x1$, fitted with <code>glm</code> . Must be of family <code>gaussian</code> , <code>binomial</code> , or <code>poisson</code> .
<code>colstoadd</code>	Number of columns in <code>xr</code> . The full model tested will be $y \sim x1 + xr$.
<code>testtype</code>	Character string specifying the test type. One of <code>"lrt"</code> (default), <code>"score"</code> , <code>"robustscore"</code> , <code>"wald"</code> , or <code>"robustwald"</code> .

Value

An object of class `lsregmem` containing pre-allocated matrices and cached quantities from the base model, for use with [addcovar](#).

Examples

```
datafile <- system.file("extdata", "simulated_data.rds", package = "lsReg")
dat <- readRDS(datafile)
basemdl <- glm(ylin ~ x1 + x2, data = dat, family = gaussian)
mem <- lsReg(basemdl, colstoadd = 1, testtype = "wald")
```

Index

addcovar, [2](#), [3](#)

lsReg, [2](#), [3](#)