# Package ‘nnsolve’

April 12, 2026

**Type** Package

**Title** Fast Non-Negative Least Squares

**Version** 0.0.2

**Date** 2026-03-25

**Description** Provides a fast algorithm for solving non-negative least squares problems. It implements the Fast Non-Negative Least Squares algorithm. of Bro and De Jong (1997)<doi:10.1002/(SICI)1099-128X(199709/10)11:53.0.CO;2-L>.

**License** GPL (>= 2)

**Imports** Rcpp, Rfast

**LinkingTo** Rcpp, RcppEigen

**RoxygenNote** 7.3.3

**Encoding** UTF-8

**NeedsCompilation** yes

**Author** Nikolaos Kontemeniotis [aut, cre],
Michail Tsagris [aut]

**Maintainer** Nikolaos Kontemeniotis <kontemeniotisn@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-04-12 16:10:02 UTC

## Contents

---

`fnnls` *Fast Non-Negative Least Squares*

---

## Description

Solves the NNLS problem min ||y - Xw||^2 subject to w >= 0 using the Fast Non-Negative Least Squares algorithm of Bro & de Jong (1997).

## Usage

```
fnnls(
  XtX,
  Xty,
  tol = 1e-06,
  max_iter = 1000,
  sum_to_constant = FALSE,
  constant = 1,
  lower_bound = FALSE,
  lb = 0
)
```

## Arguments

| | |
|---|---|
| `XtX` | A symmetric positive definite matrix of dimensions k x k. |
| `Xty` | A numeric vector of length k. |
| `tol` | The convergence tolerance, default is 1e-6. |
| `max_iter` | The maximum number of iterations, default is 1000. |
| `sum_to_constant` | If TRUE all entries sum to 'constant', Default is FALSE. |
| `constant` | If sum_to_constant is TRUE, all entries sum to this number. The default value is 1. |
| `lower_bound` | If TRUE all entries bounded below by 'lb', otherwise they are nonnegative. The default value is FALSE. |
| `lb` | If lower_bound is TRUE all entries are bounded below by 'lb'. The default value is 0. |

## Value

A non-negative numeric vector of length k with the estimated coefficients.

## References

Bro, Rasmus & Jong, Sijmen. (1997). A Fast Non-negativity-constrained Least Squares Algorithm. Journal of Chemometrics. 11. 393-401. 10.1002/(SICI)1099-128X(199709/10)11:53.0.CO;2-L.

## Examples

```
k <- 10
D <- 100
H <- matrix(rnorm(k * D), nrow = k, ncol = D)
x <- rnorm(D)
XtX <- H %*% t(H) + diag(1e-8, k)
Xty <- as.vector(H %*% x)
w <- fnnls(XtX, Xty)
```

---

# fnnls_reg    *Fast Non-Negative Least Squares Regression*

---

## Description

Solves the NNLS problem min ||y - Xb||^2 subject to b >= 0 using the Fast Non-Negative Least Squares algorithm of Bro & de Jong (1997).

## Usage

```
fnnls_reg(
  y,
  X,
  tol = 1e-06,
  max_iter = 1000,
  sum_to_constant = FALSE,
  constant = 1,
  lower_bound = FALSE,
  lb = 0
)
```

## Arguments

| | |
|---|---|
| `y` | A numeric vector of length n. |
| `X` | A numeric matrix of dimensions n x k. |
| `tol` | The convergence tolerance, default is 1e-6. |
| `max_iter` | The maximum number of iterations, default is 1000. |
| `sum_to_constant` | If TRUE all entries sum to 'constant', Default is FALSE. |
| `constant` | If sum_to_constant is TRUE, all entries sum to this number. The default value is 1. |
| `lower_bound` | If TRUE all entries bounded below by 'lb', otherwise they are nonnegative. The default value is FALSE. |
| `lb` | If lower_bound is TRUE all entries are bounded below by 'lb'. The default value is 0. |

### Value

A list with two elements:

- b: A non-negative numeric vector of length k with the estimated coefficients.
- mse: The mean squared error of the fitted model.

### References

Bro, Rasmus & Jong, Sijmen. (1997). A Fast Non-negativity-constrained Least Squares Algorithm. Journal of Chemometrics. 11. 393-401. 10.1002/(SICI)1099-128X(199709/10)11:53.0.CO;2-L.

### Examples

```
n <- 100
k <- 10
X <- matrix(rnorm(n * k), nrow = n, ncol = k)
true_b <- abs(rnorm(k))
y <- X %*% true_b + rnorm(n, sd = 0.1)
result <- fnnls_reg(y, X)
result$b
result$mse
```

---

## fnnls_regs — *Fast Non-Negative Least Squares for Multiple Outputs*

---

### Description

Solves the NNLS problem min ||Y - XB||_F^2 subject to B >= 0 using the Fast Non-Negative Least Squares algorithm of Bro & de Jong (1997).

### Usage

```
fnnls_regs(
  Y,
  X,
  tol = 1e-06,
  max_iter = 1000,
  sum_to_constant = FALSE,
  constant = 1,
  lower_bound = FALSE,
  lb = 0,
  parallel = FALSE,
  ncores = -1
)
```

### Arguments

| | |
|---|---|
| `Y` | A numeric matrix of dimensions n x m. |
| `X` | A numeric matrix of dimensions n x p. |
| `tol` | The convergence tolerance, default is 1e-6. |
| `max_iter` | The maximum number of iterations, default is 1000. |
| `sum_to_constant` | If TRUE all entries in each column of B sum to 'constant'. Default is FALSE. |
| `constant` | If sum_to_constant is TRUE, all entries in each column sum to this number. The default value is 1. |
| `lower_bound` | If TRUE all entries bounded below by 'lb', otherwise they are nonnegative. The default value is FALSE. |
| `lb` | If lower_bound is TRUE all entries are bounded below by 'lb'. The default value is 0. |
| `parallel` | If TRUE, the columns of B are computed in parallel. The default value is FALSE. |
| `ncores` | If parallel is TRUE, this many cores are used in the parallel computations. Must be positive integer. The default value is -1 (use all available cores). |

### Value

A list with two elements:

- `B`: A non-negative numeric matrix of dimensions p x m with the estimated coefficients.
- `mse`: A numeric vector of length m with the mean squared error for each output column.

### References

Bro, Rasmus & Jong, Sijmen. (1997). A Fast Non-negativity-constrained Least Squares Algorithm. Journal of Chemometrics. 11. 393-401. 10.1002/(SICI)1099-128X(199709/10)11:53.0.CO;2-L.

### Examples

```
n <- 50
p <- 10
m <- 3
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
Y <- matrix(runif(n * m, min = 0, max = 10), nrow = n, ncol = m)
result <- fnnls_regs(Y, X, tol = 1e-8, max_iter = 1000)
result$B
result$mse
```

# Index