Package 'sparseMatrixStats'

October 17, 2020

Type Package

Title Summary Statistics for Rows and Columns of Sparse Matrices

Version 1.0.5

Description High performance functions for row and column operations on sparse matrices. For example: col / rowMeans2, col / rowMedians, col / rowVars etc. Currently, the optimizations are limited to data in the column sparse format. This package is inspired by the matrixStats package by Henrik Bengtsson.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

LinkingTo Rcpp

Imports Rcpp, Matrix, MatrixGenerics, matrixStats, methods

Suggests testthat (>= 2.1.0), knitr, bench, rmarkdown, BiocStyle

SystemRequirements C++14

VignetteBuilder knitr

biocViews Infrastructure, Software, DataRepresentation

git_url https://git.bioconductor.org/packages/sparseMatrixStats

git_branch RELEASE_3_11

git_last_commit 0eaf6f7

git_last_commit_date 2020-05-24

Date/Publication 2020-10-16

Author Constantin Ahlmann-Eltze [aut, cre] (<https://orcid.org/0000-0002-3762-068X>)

Maintainer Constantin Ahlmann-Eltze <artjom31415@googlemail.com>

R topics documented:

`colAlls,dgCMatrix-method`

Check if all elements in a row (column) of a matrix-like object are equal to a value

---

Description

Check if all elements in a row (column) of a matrix-like object are equal to a value.

Usage

```
## S4 method for signature 'dgCMatrix'
colAlls(x, rows = NULL, cols = NULL, value = TRUE, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowAlls(x, rows = NULL, cols = NULL, value = TRUE, na.rm = FALSE)
```

Arguments

x	An NxK matrix-like object.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
value	The value to search for.
na.rm	If TRUE, NAs are excluded first, otherwise not.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowAlls / matrixStats::colAlls.

Value

Returns a logical vector of length N (K).

See Also

- matrixStats::rowAlls() and matrixStats::colAlls() which are used when the input is a matrix, array, or numeric vector.
- For checks if *any* element is equal to a value, see rowAnys().
- base::all().

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowAlls(mat)
  colAlls(mat)
```

---

colAnyNAs,dgCMatrix-method

Check if any elements in a row (column) of a matrix-like object is missing

---

Description

Check if any elements in a row (column) of a matrix-like object is missing.

Usage

```
## S4 method for signature 'dgCMatrix'
colAnyNAs(x, rows = NULL, cols = NULL)

## S4 method for signature 'dgCMatrix'
rowAnyNAs(x, rows = NULL, cols = NULL)
```

Arguments

x	An NxK matrix-like object.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowAnyNAs / matrixStats::colAnyNAs.

Value

Returns a logical vector of length N (K).

See Also

- matrixStats::rowAnyNAs() and matrixStats::colAnyNAs() which are used when the input is a matrix or numeric vector.
- For checks if any element is equal to a value, see rowAnys().
- base::is.na() and base::any().

Examples

```
mat <- matrix(0, nrow=10, ncol=5)
mat[sample(seq_len(5 *10), 5)] <- NA
sp_mat <- as(mat, "dgCMatrix")
colAnyNAs(sp_mat)
```

---

colAnys,dgCMatrix-method

Check if any elements in a row (column) of a matrix-like object is equal to a value

---

Description

Check if any elements in a row (column) of a matrix-like object is equal to a value.

Usage

```
## S4 method for signature 'dgCMatrix'
colAnys(x, rows = NULL, cols = NULL, value = TRUE, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowAnys(x, rows = NULL, cols = NULL, value = TRUE, na.rm = FALSE)
```

Arguments

x	An NxK matrix-like object.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
value	The value to search for.
na.rm	If TRUE, NAs are excluded first, otherwise not.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowAnys / matrixStats::colAnys.

Value

Returns a logical vector of length N (K).

See Also

- matrixStats::rowAnys() and matrixStats::colAnys() which are used when the input is a matrix or numeric vector.
- base::any().

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowAnys(mat)
  colAnys(mat)
```

---

colAvgsPerRowSet,dgCMatrix-method

Calculates for each row (column) a summary statistic for equally sized subsets of columns (rows)

Description

Calculates for each row (column) a summary statistic for equally sized subsets of columns (rows)

Usage

```
## S4 method for signature 'dgCMatrix'
colAvgsPerRowSet(
  X,
  W = NULL,
  cols = NULL,
  S,
  FUN = colMeans2,
  ...,
  tFUN = FALSE
)

## S4 method for signature 'dgCMatrix'
rowAvgsPerColSet(
  X,
  W = NULL,
  rows = NULL,
  S,
  FUN = rowMeans2,
  ...,
  tFUN = FALSE
)
```

Arguments

X	An NxM matrix-like object.
W	An optional numeric NxM matrix of weights.
cols	A vector indicating the subset (and/or columns) to operate over. If NULL, no subsetting is done.
S	An integer KxJ matrix that specifying the J subsets. Each column hold K column (row) indices for the corresponding subset. The range of values is [1, M] ([1,N]).
FUN	A row-by-row (column-by-column) summary statistic function. It is applied to to each column (row) subset of X that is specified by S.
...	Additional arguments passed to FUN.
tFUN	If TRUE, X is transposed before it is passed to FUN.
rows	A vector indicating the subset (and/or columns) to operate over. If NULL, no subsetting is done.

Details

Note: the handling of missing parameters differs from [matrixStats::colAvgsPerRowSet()]. The ‘matrixStats‘ version always removes ‘NA‘’s if there are any in the data. This method however does whatever is passed in the ‘...‘ parameter.

Value

Returns a numeric JxN (MxJ) matrix.

See Also

- matrixStats::rowAvgsPerColSet() and matrixStats::colAvgsPerRowSet() which are used when the input is a matrix or numeric vector.

Examples

```
mat <- matrix(rnorm(20), nrow = 5, ncol = 4)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  S <- matrix(1:ncol(mat), ncol = 2)
  print(S)

  rowAvgsPerColSet(mat, S = S, FUN = rowMeans)
  rowAvgsPerColSet(mat, S = S, FUN = rowVars)
```

---

colCollapse,dgCMatrix-method

Extract one cell from each row (column) of a matrix-like object

---

Description

Extract one cell from each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colCollapse(x, idxs, cols = NULL)

## S4 method for signature 'dgCMatrix'
rowCollapse(x, idxs, rows = NULL)
```

Arguments

x	An NxK matrix-like object.
idxs	An index vector with the position to extract. It is recycled to match the number of rows (column)
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowCollapse / matrixStats::colCollapse.

Value

Returns a numeric vector of length N (K).

See Also

- matrixStats::rowCollapse() and matrixStats::colCollapse() which are used when the input is a matrix or numeric vector.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowCollapse(mat, idxs = 2)
  rowCollapse(mat, idxs = c(1,1,2,3,2))

  colCollapse (mat, idxs = 4)
```

---

colCounts,dgCMatrix-method

Count how often an element in a row (column) of a matrix-like object is equal to a value

---

Description

Count how often an element in a row (column) of a matrix-like object is equal to a value.

Usage

```
## S4 method for signature 'dgCMatrix'
colCounts(x, rows = NULL, cols = NULL, value = TRUE, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowCounts(x, rows = NULL, cols = NULL, value = TRUE, na.rm = FALSE)
```

Arguments

x	An NxK matrix-like object.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
value	The value to search for.
na.rm	If TRUE, NAs are excluded first, otherwise not.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowCounts / matrixStats::colCounts.

Value

Returns a integer vector of length N (K).

See Also

- matrixStats::rowCounts() and matrixStats::colCounts() which are used when the input is a matrix or numeric vector.
- For checks if any element is equal to a value, see rowAnys(). To check if all elements are equal, see rowAlls().

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowCounts(mat)
  colCounts(mat)

  rowCounts(mat, value = 0)
  colCounts(mat, value = Inf, na.rm = TRUE)
```

---

colCummaxs,dgCMatrix-method

Calculates the cumulative maxima for each row (column) of a matrix-like object

---

Description

Calculates the cumulative maxima for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colCummaxs(x, rows = NULL, cols = NULL)

## S4 method for signature 'dgCMatrix'
rowCummaxs(x, rows = NULL, cols = NULL)
```

Arguments

x	An NxK matrix-like object.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowCummaxs / matrixStats::colCummaxs.

Value

Returns a numeric matrixwith the same dimensions as x.

See Also

- matrixStats::rowCummaxs() and matrixStats::colCummaxs() which are used when the input is a matrix or numeric vector.
- For single maximum estimates, see rowMaxs().
- base::cummax().

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowCummaxs(mat)
  colCummaxs(mat)
```

---

colCummins,dgCMatrix-method

Calculates the cumulative minima for each row (column) of a matrix-like object

---

Description

Calculates the cumulative minima for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colCummins(x, rows = NULL, cols = NULL)

## S4 method for signature 'dgCMatrix'
rowCummins(x, rows = NULL, cols = NULL)
```

Arguments

x	An NxK matrix-like object.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowCummins / matrixStats::colCummins.

Value

Returns a numeric matrixwith the same dimensions as x.

See Also

- matrixStats::rowCummins() and matrixStats::colCummins() which are used when the input is a matrix or numeric vector.
- For single minimum estimates, see rowMins().
- base::cummin().

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowCummins(mat)
  colCummins(mat)
```

---

colCumprods,dgCMatrix-method

Calculates the cumulative product for each row (column) of a matrix-like object

---

Description

Calculates the cumulative product for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colCumprods(x, rows = NULL, cols = NULL)

## S4 method for signature 'dgCMatrix'
rowCumprods(x, rows = NULL, cols = NULL)
```

Arguments

x	An NxK matrix-like object.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowCumprods / matrixStats::colCumprods.

Value

Returns a numeric matrixwith the same dimensions as x.

See Also

- matrixStats::rowCumprods() and matrixStats::colCumprods() which are used when the input is a matrix or numeric vector.
- base::cumprod().

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowCumprods(mat)
  colCumprods(mat)
```

---

colCumsums,dgCMatrix-method

Calculates the cumulative sum for each row (column) of a matrix-like object

---

Description

Calculates the cumulative sum for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colCumsums(x, rows = NULL, cols = NULL)

## S4 method for signature 'dgCMatrix'
rowCumsums(x, rows = NULL, cols = NULL)
```

Arguments

x	An NxK matrix-like object.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowCumsums / matrixStats::colCumsums.

Value

Returns a numeric matrixwith the same dimensions as x.

See Also

- matrixStats::rowCumsums() and matrixStats::colCumsums() which are used when the input is a matrix or numeric vector.
- base::cumsum().

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowCumsums(mat)
  colCumsums(mat)
```

---

colDiffs,dgCMatrix-method

Calculates the difference between each element of a row (column) of a matrix-like object

---

Description

Calculates the difference between each element of a row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colDiffs(x, rows = NULL, cols = NULL, lag = 1L, differences = 1L)

## S4 method for signature 'dgCMatrix'
rowDiffs(x, rows = NULL, cols = NULL, lag = 1L, differences = 1L)
```

Arguments

x	An NxK matrix-like object.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
lag	An integer specifying the lag.
differences	An integer specifying the order of difference.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowDiffs / matrixStats::colDiffs.

Value

Returns a numeric matrix with one column (row) less than x: $Nx(K-1)$ or $(N-1)xK$.

See Also

- matrixStats::rowDiffs() and matrixStats::colDiffs() which are used when the input is a matrix or numeric vector.
- base::diff().

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowDiffs(mat)
  colDiffs(mat)
```

---

colIQRDiffs,dgCMatrix-method

Calculates the interquartile range of the difference between each element of a row (column) of a matrix-like object

---

Description

Calculates the interquartile range of the difference between each element of a row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colIQRDiffs(x, rows = NULL, cols = NULL, na.rm = FALSE, diff = 1L, trim = 0)

## S4 method for signature 'dgCMatrix'
rowIQRDiffs(x, rows = NULL, cols = NULL, na.rm = FALSE, diff = 1L, trim = 0)
```

Arguments

x	An NxK matrix-like object.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
na.rm	If TRUE, NAs are excluded first, otherwise not.
diff	An integer specifying the order of difference.
trim	A double in [0,1/2] specifying the fraction of observations to be trimmed from each end of (sorted) x before estimation.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowIQRDiffs / matrixStats::colIQRDiffs.

Value

Returns a numeric vector of length N (K).

See Also

- matrixStats::rowIQRDiffs() and matrixStats::colIQRDiffs() which are used when the input is a matrix or numeric vector.
- For the direct interquartile range see also rowIQRs.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowIQRDiffs(mat)
  colIQRDiffs(mat)
```

---

colIQRs,dgCMatrix-method

Calculates the interquartile range for each row (column) of a matrix-like object

---

Description

Calculates the interquartile range for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colIQRs(x, rows = NULL, cols = NULL, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowIQRs(x, rows = NULL, cols = NULL, na.rm = FALSE)
```

Arguments

x	An NxK matrix-like object.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
na.rm	If TRUE, NAs are excluded first, otherwise not.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowIQRs / matrixStats::colIQRs.

Value

Returns a numeric vector of length N (K).

See Also

- matrixStats::rowIQRs() and matrixStats::colIQRs() which are used when the input is a matrix or numeric vector.
- For a non-robust analog, see rowSds(). For a more robust version see rowMads()
- stats::IQR().

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowIQRs(mat)
  colIQRs(mat)
```

---

colLogSumExps,dgCMatrix-method

Accurately calculates the logarithm of the sum of exponentials for each row (column) of a matrix-like object

---

Description

Accurately calculates the logarithm of the sum of exponentials for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colLogSumExps(lx, rows = NULL, cols = NULL, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowLogSumExps(lx, rows = NULL, cols = NULL, na.rm = FALSE)
```

Arguments

lx	An NxK matrix-like object. Typically lx are log(x) values.
rows	A vector indicating the subset (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset (and/or columns) to operate over. If NULL, no subsetting is done.
na.rm	If TRUE, NAs are excluded first, otherwise not.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowLogSumExps / matrixStats::colLogSumExps.

Value

Returns a numeric vector of length N (K).

See Also

- matrixStats::rowLogSumExps() and matrixStats::colLogSumExps() which are used when the input is a matrix or numeric vector.
- rowSums2()

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowLogSumExps(mat)
  colLogSumExps(mat)
```

---

colMadDiffs,dgCMatrix-method

Calculates the mean absolute deviation of the difference between each element of a row (column) of a matrix-like object

---

Description

Calculates the mean absolute deviation of the difference between each element of a row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colMadDiffs(
  x,
  rows = NULL,
  cols = NULL,
  na.rm = FALSE,
  diff = 1L,
  trim = 0,
  constant = 1.4826
)

## S4 method for signature 'dgCMatrix'
rowMadDiffs(
```

```
  x,
  rows = NULL,
  cols = NULL,
  na.rm = FALSE,
  diff = 1L,
  trim = 0,
  constant = 1.4826
)
```

Arguments

x	An NxK matrix-like object.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
na.rm	If TRUE, NAs are excluded first, otherwise not.
diff	An integer specifying the order of difference.
trim	A double in [0,1/2] specifying the fraction of observations to be trimmed from each end of (sorted) x before estimation.
constant	A scale factor. See mad for details.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowMadDiffs / matrixStats::colMadDiffs.

Value

Returns a numeric vector of length N (K).

See Also

- matrixStats::rowMadDiffs() and matrixStats::colMadDiffs() which are used when the input is a matrix or numeric vector.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowMadDiffs(mat)
  colMadDiffs(mat)
```

`colMads,dgCMatrix-method` *Calculates the median absolute deviation for each row (column) of a matrix-like object*

Description

Calculates the median absolute deviation for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colMads(x, rows = NULL, cols = NULL, constant = 1.4826, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowMads(x, rows = NULL, cols = NULL, constant = 1.4826, na.rm = FALSE)
```

Arguments

`x`	An NxK matrix-like object.
`rows`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`cols`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`constant`	A scale factor. See `stats::mad()` for details.
`na.rm`	If `TRUE`, `NA`s are excluded first, otherwise not.

Details

The S4 methods for `x` of type `matrix`, `array`, or `numeric` call `matrixStats::rowMads` / `matrixStats::colMads`.

Value

Returns a `numeric vector` of length N (K).

See Also

- `matrixStats::rowMads()` and `matrixStats::colMads()` which are used when the input is a `matrix` or `numeric` vector.
- For mean estimates, see `rowMeans2()` and `rowMeans()`.
- For non-robust standard deviation estimates, see `rowSds()`.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)
```

```
rowMads(mat)
colMads(mat)
```

`colMaxs,dgCMatrix-method`

Calculates the maximum for each row (column) of a matrix-like object

---

Description

Calculates the maximum for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colMaxs(x, rows = NULL, cols = NULL, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowMaxs(x, rows = NULL, cols = NULL, na.rm = FALSE)
```

Arguments

`x`	An NxK matrix-like object.
`rows`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`cols`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`na.rm`	If `TRUE`, `NA`s are excluded first, otherwise not.

Details

The S4 methods for `x` of type `matrix`, `array`, or `numeric` call `matrixStats::rowMaxs` / `matrixStats::colMaxs`.

Value

Returns a `numeric` `vector` of length N (K).

See Also

- `matrixStats::rowMaxs()` and `matrixStats::colMaxs()` which are used when the input is a `matrix` or `numeric` vector.
- For min estimates, see `rowMins()`.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)
```

```
rowMaxs(mat)
colMaxs(mat)
```

`colMeans2,dgCMatrix-method`

Calculates the mean for each row (column) of a matrix-like object

---

Description

Calculates the mean for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colMeans2(x, rows = NULL, cols = NULL, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowMeans2(x, rows = NULL, cols = NULL, na.rm = FALSE)
```

Arguments

`x`	An NxK matrix-like object.
`rows`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`cols`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`na.rm`	If `TRUE`, `NA`s are excluded first, otherwise not.

Details

The S4 methods for x of type `matrix`, `array`, or `numeric` call `matrixStats::rowMeans2` / `matrixStats::colMeans2`.

Value

Returns a `numeric` `vector` of length N (K).

See Also

- `matrixStats::rowMeans2()` and `matrixStats::colMeans2()` which are used when the input is a `matrix` or `numeric` vector.
- See also `rowMeans()` for the corresponding function in base R.
- For variance estimates, see `rowVars()`.
- See also the base R version `base::rowMeans()`.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowMeans2(mat)
  colMeans2(mat)
```

`colMedians,dgCMatrix-method`
Calculates the median for each row (column) of a matrix-like object

---

Description

Calculates the median for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colMedians(x, rows = NULL, cols = NULL, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowMedians(x, rows = NULL, cols = NULL, na.rm = FALSE)
```

Arguments

`x`	An NxK matrix-like object.
`rows`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`cols`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`na.rm`	If `TRUE`, `NA`s are excluded first, otherwise not.

Details

The S4 methods for `x` of type `matrix`, `array`, or `numeric` call `matrixStats::rowMedians` / `matrixStats::colMedians`.

Value

Returns a `numeric` `vector` of length N (K).

See Also

- `matrixStats::rowMedians()` and `matrixStats::colMedians()` which are used when the input is a `matrix` or `numeric` vector.
- For mean estimates, see `rowMeans2()` and `rowMeans()`.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowMedians(mat)
  colMedians(mat)
```

`colMins,dgCMatrix-method`

Calculates the minimum for each row (column) of a matrix-like object

---

Description

Calculates the minimum for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colMins(x, rows = NULL, cols = NULL, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowMins(x, rows = NULL, cols = NULL, na.rm = FALSE)
```

Arguments

`x`	An NxK matrix-like object.
`rows`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`cols`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`na.rm`	If `TRUE`, `NA`s are excluded first, otherwise not.

Details

The S4 methods for `x` of type `matrix`, `array`, or `numeric` call `matrixStats::rowMins` / `matrixStats::colMins`.

Value

Returns a `numeric` `vector` of length N (K).

See Also

- `matrixStats::rowMins()` and `matrixStats::colMins()` which are used when the input is a `matrix` or `numeric` vector.
- For max estimates, see `rowMaxs()`.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowMins(mat)
  colMins(mat)
```

`colOrderStats,dgCMatrix-method`
Calculates an order statistic for each row (column) of a matrix-like object

---

Description

Calculates an order statistic for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colOrderStats(x, rows = NULL, cols = NULL, which = 1, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowOrderStats(x, rows = NULL, cols = NULL, which = 1, na.rm = FALSE)
```

Arguments

`x`	An NxK matrix-like object.
`rows`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`cols`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`which`	An integer index in [1,K] ([1,N]) indicating which order statistic to be returned
`na.rm`	If TRUE, NAs are excluded first, otherwise not.

Details

The S4 methods for `x` of type `matrix`, `array`, or `numeric` call `matrixStats::rowOrderStats` / `matrixStats::colOrderStats`.

Value

Returns a `numeric vector` of length N (K).

See Also

- `matrixStats::rowOrderStats()` and `matrixStats::colOrderStats()` which are used when the input is a `matrix` or `numeric` vector.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
    mat[2, 1] <- 2
    mat[3, 3] <- Inf
    mat[4, 1] <- 0

    print(mat)

    rowOrderStats(mat, which = 1)
    colOrderStats(mat, which = 3)
```

`colProds,dgCMatrix-method`

Calculates the product for each row (column) in a matrix

Description

Calculates the product for each row (column) in a matrix

Usage

```
## S4 method for signature 'dgCMatrix'
colProds(x, rows = NULL, cols = NULL, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowProds(x, rows = NULL, cols = NULL, na.rm = FALSE)
```

Arguments

`x`	An NxK matrix-like object.
`rows`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`cols`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`na.rm`	If `TRUE`, `NA`s are excluded first, otherwise not.

Details

Attention: This method ignores the order of the values, because it assumes that the product is commutative. Unfortunately, for 'double' this is not true. For example 'NaN * NA = NaN', but 'NA * NaN = NA'. This is relevant for this function if there are '+-Inf', because 'Inf * 0 = NaN'. This function returns 'NA' whenever there is 'NA' in the input. This is different from 'matrixStats::colProds()'.

Value

Returns a `numeric` `vector` of length N (K).

See Also

- `matrixStats::rowProds()` and `matrixStats::colProds()` which are used when the input is a `matrix` or `numeric` vector.
- For sums across rows (columns), see `rowSums2()` (`colSums2()`)
- `base::prod()`.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowProds(mat)
  colProds(mat)
```

---

colQuantiles,dgCMatrix-method

Calculates quantiles for each row (column) of a matrix-like object

---

Description

Calculates quantiles for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colQuantiles(
  x,
  rows = NULL,
  cols = NULL,
  probs = seq(from = 0, to = 1, by = 0.25),
  na.rm = FALSE,
  drop = TRUE
)

## S4 method for signature 'dgCMatrix'
rowQuantiles(
  x,
  rows = NULL,
  cols = NULL,
  probs = seq(from = 0, to = 1, by = 0.25),
  na.rm = FALSE,
  drop = TRUE
)
```

Arguments

x	An NxK matrix-like object.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
probs	A numeric vector of J probabilities in [0, 1].
na.rm	If TRUE, NAs are excluded first, otherwise not.
drop	If TRUE a vector is returned if J == 1. Note, that this is not a generic argument and not all implementation of this function have to provide it.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowQuantiles / matrixStats::colQuantiles.

Value

a numeric NxJ (KxJ) matrix, where N (K) is the number of rows (columns) for which the J values are calculated.

See Also

- matrixStats::rowQuantiles() and matrixStats::colQuantiles() which are used when the input is a matrix or numeric vector.
- stats::quantile

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowQuantiles(mat)
  colQuantiles(mat)
```

---

colRanges,dgCMatrix-method

Calculates the minimum and maximum for each row (column) of a matrix-like object

---

Description

Calculates the minimum and maximum for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colRanges(x, rows = NULL, cols = NULL, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowRanges(x, rows = NULL, cols = NULL, na.rm = FALSE)
```

Arguments

x	An NxK matrix-like object.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
na.rm	If TRUE, NAs are excluded first, otherwise not.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowRanges / matrixStats::colRanges.

Value

a numeric Nx2 (Kx2) matrix, where N (K) is the number of rows (columns) for which the ranges are calculated.

See Also

- matrixStats::rowRanges() and matrixStats::colRanges() which are used when the input is a matrix or numeric vector.
- For max estimates, see rowMaxs().
- For min estimates, see rowMins().
- base::range().

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowRanges(mat)
  colRanges(mat)
```

`colRanks,dgCMatrix-method` *Calculates the rank of the elements for each row (column) of a matrix-like object*

Description

Calculates the rank of the elements for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colRanks(
  x,
  rows = NULL,
  cols = NULL,
  ties.method = c("max", "average", "min"),
  preserve.shape = FALSE,
  na.handling = c("keep", "last")
)

## S4 method for signature 'dgCMatrix'
rowRanks(
  x,
  rows = NULL,
  cols = NULL,
  ties.method = c("max", "average", "min"),
  preserve.shape = TRUE,
  na.handling = c("keep", "last")
)
```

Arguments

`x`	An NxK matrix-like object.
`rows`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`cols`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`ties.method`	A character string specifying how ties are treated. Note that the default specifies fewer options than the original matrixStats package.
`preserve.shape`	a boolean that specifies if the returned matrix has the same dimensions as the input matrix. By default this is true for 'rowRanks()', but false for 'colRanks()'.
`na.handling`	string specifying how 'NA's are handled. They can either be preserved with an 'NA' rank ('keep') or sorted in at the end ('last'). Default is 'keep' derived from the behavior of the equivalent

Details

There are three different methods available for handling ties:

'max' for values with identical values the maximum rank is returned

'average' for values with identical values the average of the ranks they cover is returned. Note, that in this case the return value is of type 'numeric'.

'min' for values with identical values the minimum rank is returned.

Value

a matrix of type `integer` is returned unless `ties.method = "average"`. Ithas dimensions' `NxJ` (`KxJ`) `matrix`, where N (K) is the number of rows (columns) of the input x.

See Also

- `matrixStats::rowRanks()` and `matrixStats::colRanks()` which are used when the input is a `matrix` or `numeric` vector.
- base::rank

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowRanks(mat)
  colRanks(mat)
```

---

colSdDiffs,dgCMatrix-method

Calculates the standard deviation of the difference between each element of a row (column) of a matrix-like object

---

Description

Calculates the standard deviation of the difference between each element of a row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colSdDiffs(x, rows = NULL, cols = NULL, na.rm = FALSE, diff = 1L, trim = 0)

## S4 method for signature 'dgCMatrix'
rowSdDiffs(x, rows = NULL, cols = NULL, na.rm = FALSE, diff = 1L, trim = 0)
```

Arguments

x	An NxK matrix-like object.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
na.rm	If TRUE, NAs are excluded first, otherwise not.
diff	An integer specifying the order of difference.
trim	A double in [0,1/2] specifying the fraction of observations to be trimmed from each end of (sorted) x before estimation.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowSdDiffs / matrixStats::colSdDiffs.

Value

Returns a numeric vector of length N (K).

See Also

- matrixStats::rowSdDiffs() and matrixStats::colSdDiffs() which are used when the input is a matrix or numeric vector.
- for the direct standard deviation see rowSds().

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowSdDiffs(mat)
  colSdDiffs(mat)
```

---

colSds,dgCMatrix-method

Calculates the standard deviation for each row (column) of a matrix-like object

---

Description

Calculates the standard deviation for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colSds(x, rows = NULL, cols = NULL, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowSds(x, rows = NULL, cols = NULL, na.rm = FALSE)
```

Arguments

x	An NxK matrix-like object.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
na.rm	If TRUE, NAs are excluded first, otherwise not.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowSds / matrixStats::colSds.

Value

Returns a numeric vector of length N (K).

See Also

- matrixStats::rowSds() and matrixStats::colSds() which are used when the input is a matrix or numeric vector.
- For mean estimates, see rowMeans2() and rowMeans().
- For variance estimates, see rowVars().

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowSds(mat)
  colSds(mat)
```

`colSums2,dgCMatrix-method` *Calculates the sum for each row (column) of a matrix-like object*

Description

Calculates the sum for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colSums2(x, rows = NULL, cols = NULL, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowSums2(x, rows = NULL, cols = NULL, na.rm = FALSE)
```

Arguments

`x`	An NxK matrix-like object.
`rows`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`cols`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`na.rm`	If `TRUE`, `NA`s are excluded first, otherwise not.

Details

The S4 methods for x of type `matrix`, `array`, or `numeric` call `matrixStats::rowSums2` / `matrixStats::colSums2`.

Value

Returns a `numeric` `vector` of length N (K).

See Also

- `matrixStats::rowSums2()` and `matrixStats::colSums2()` which are used when the input is a `matrix` or `numeric` vector.
- For mean estimates, see `rowMeans2()` and `rowMeans()`.
- `base::sum()`.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowSums2(mat)
  colSums2(mat)
```

`colTabulates,dgCMatrix-method`
Tabulates the values in a matrix-like object by row (column)

Description

Tabulates the values in a matrix-like object by row (column).

Usage

```
## S4 method for signature 'dgCMatrix'
colTabulates(x, rows = NULL, cols = NULL, values = NULL)

## S4 method for signature 'dgCMatrix'
rowTabulates(x, rows = NULL, cols = NULL, values = NULL)
```

Arguments

`x`	An NxK matrix-like object.
`rows`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`cols`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`values`	the values to search for.

Details

The S4 methods for `x` of type `matrix`, `array`, or `numeric` call `matrixStats::rowTabulates` / `matrixStats::colTabulates`.

Value

a `numeric` NxJ (KxJ) `matrix`, where N (K) is the number of rows (columns) for which the J values are calculated.

See Also

- `matrixStats::rowTabulates()` and `matrixStats::colTabulates()` which are used when the input is a `matrix` or `numeric` vector.
- `base::table()`

Examples

```
mat <- matrix(rpois(15, lambda = 3), nrow = 5, ncol = 3)
  mat[2, 1] <- NA_integer_
  mat[3, 3] <- 0L
  mat[4, 1] <- 0L

  print(mat)

  rowTabulates(mat)
```

```
colTabulates(mat)

rowTabulates(mat, values = 0)
colTabulates(mat, values = 0)
```

`colVarDiffs,dgCMatrix-method`

Calculates the variance of the difference between each element of a row (column) of a matrix-like object

---

Description

Calculates the variance of the difference between each element of a row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colVarDiffs(x, rows = NULL, cols = NULL, na.rm = FALSE, diff = 1L, trim = 0)

## S4 method for signature 'dgCMatrix'
rowVarDiffs(x, rows = NULL, cols = NULL, na.rm = FALSE, diff = 1L, trim = 0)
```

Arguments

`x`	An NxK matrix-like object.
`rows`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`cols`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`na.rm`	If `TRUE`, `NA`s are excluded first, otherwise not.
`diff`	An integer specifying the order of difference.
`trim`	A double in [0,1/2] specifying the fraction of observations to be trimmed from each end of (sorted) x before estimation.

Details

The S4 methods for `x` of type `matrix`, `array`, or `numeric` call `matrixStats::rowVarDiffs` / `matrixStats::colVarDiffs`.

Value

Returns a `numeric` `vector` of length N (K).

See Also

- `matrixStats::rowVarDiffs()` and `matrixStats::colVarDiffs()` which are used when the input is a `matrix` or `numeric` vector.
- for the direct variance see `rowVars()`.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowVarDiffs(mat)
  colVarDiffs(mat)
```

`colVars,dgCMatrix-method`

Calculates the variance for each row (column) of a matrix-like object

---

Description

Calculates the variance for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colVars(x, rows = NULL, cols = NULL, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowVars(x, rows = NULL, cols = NULL, na.rm = FALSE)
```

Arguments

`x`	An NxK matrix-like object.
`rows`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`cols`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`na.rm`	If `TRUE`, `NA`s are excluded first, otherwise not.

Details

The S4 methods for `x` of type `matrix`, `array`, or `numeric` call `matrixStats::rowVars` / `matrixStats::colVars`.

Value

Returns a `numeric` `vector` of length N (K).

See Also

- `matrixStats::rowVars()` and `matrixStats::colVars()` which are used when the input is a `matrix` or `numeric` vector.
- For mean estimates, see `rowMeans2()` and `rowMeans()`.
- For standard deviation estimates, see `rowSds()`.
- `stats::var()`.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)

  rowVars(mat)
  colVars(mat)
```

---

colWeightedMads,dgCMatrix-method

Calculates the weighted median absolute deviation for each row (column) of a matrix-like object

---

Description

Calculates the weighted median absolute deviation for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colWeightedMads(
  x,
  w = NULL,
  rows = NULL,
  cols = NULL,
  na.rm = FALSE,
  constant = 1.4826,
  center = NULL
)

## S4 method for signature 'dgCMatrix'
rowWeightedMads(
  x,
  w = NULL,
  rows = NULL,
  cols = NULL,
  na.rm = FALSE,
  constant = 1.4826
)
```

Arguments

x	An NxK matrix-like object.
w	A numeric vector of length K (N) that specifies by how much each element is weighted.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.

cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
na.rm	If TRUE, NAs are excluded first, otherwise not.
constant	A scale factor. See stats::mad() for details.
center	Not supported at the moment.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowWeightedMads / matrixStats::colWeightedMads.

Value

Returns a numeric vector of length N (K).

See Also

- matrixStats::rowWeightedMads() and matrixStats::colWeightedMads() which are used when the input is a matrix or numeric vector.
- See also rowMads for the corresponding unweighted function.

Examples

```
mat <- matrix(0, nrow=10, ncol=5)
mat[sample(prod(dim(mat)), 25)] <- rpois(n=25, 5)
sp_mat <- as(mat, "dgCMatrix")
weights <- rnorm(10, mean=1, sd=0.1)

# sparse version
sparseMatrixStats::colWeightedMads(sp_mat, weights)

# Attention the result differs from matrixStats
# because it always uses 'interpolate=FALSE'.
matrixStats::colWeightedMads(mat, weights)
```

---

colWeightedMeans,dgCMatrix-method

Calculates the weighted mean for each row (column) of a matrix-like object

---

Description

Calculates the weighted mean for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colWeightedMeans(x, w = NULL, rows = NULL, cols = NULL, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowWeightedMeans(x, w = NULL, rows = NULL, cols = NULL, na.rm = FALSE)
```

Arguments

x	An NxK matrix-like object.
w	A numeric vector of length K (N) that specifies by how much each element is weighted.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
na.rm	If TRUE, NAs are excluded first, otherwise not.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowWeightedMeans / matrixStats::colWeightedMeans.

Value

Returns a numeric vector of length N (K).

See Also

- matrixStats::rowWeightedMeans() and matrixStats::colWeightedMeans() which are used when the input is a matrix or numeric vector.
- See also rowMeans2 for the corresponding unweighted function.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)
  w <- rnorm(n = 5, mean = 3)
  rowWeightedMeans(mat, w = w[1:3])
  colWeightedMeans(mat, w = w)
```

---

colWeightedMedians,dgCMatrix-method

Calculates the weighted median for each row (column) of a matrix-like object

---

Description

Calculates the weighted median for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colWeightedMedians(x, w = NULL, rows = NULL, cols = NULL, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowWeightedMedians(x, w = NULL, rows = NULL, cols = NULL, na.rm = FALSE)
```

Arguments

x	An NxK matrix-like object.
w	A numeric vector of length K (N) that specifies by how much each element is weighted.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
na.rm	If TRUE, NAs are excluded first, otherwise not.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowWeightedMedians / matrixStats::colWeightedMedians.

Value

Returns a numeric vector of length N (K).

See Also

- matrixStats::rowWeightedMedians() and matrixStats::colWeightedMedians() which are used when the input is a matrix or numeric vector.
- See also rowMedians for the corresponding unweighted function.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)
  w <- rnorm(n = 5, mean = 3)
  rowWeightedMedians(mat, w = w[1:3])
  colWeightedMedians(mat, w = w)
```

colWeightedSds,dgCMatrix-method

Calculates the weighted standard deviation for each row (column) of a matrix-like object

Description

Calculates the weighted standard deviation for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colWeightedSds(x, w = NULL, rows = NULL, cols = NULL, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowWeightedSds(x, w = NULL, rows = NULL, cols = NULL, na.rm = FALSE)
```

Arguments

x	An NxK matrix-like object.
w	A numeric vector of length K (N) that specifies by how much each element is weighted.
rows	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
na.rm	If TRUE, NAs are excluded first, otherwise not.

Details

The S4 methods for x of type matrix, array, or numeric call matrixStats::rowWeightedSds / matrixStats::colWeightedSds.

Value

Returns a numeric vector of length N (K).

See Also

- matrixStats::rowWeightedSds() and matrixStats::colWeightedSds() which are used when the input is a matrix or numeric vector.
- See also rowSds for the corresponding unweighted function.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)
```

```
w <- rnorm(n = 5, mean = 3)
rowWeightedSds(mat, w = w[1:3])
colWeightedSds(mat, w = w)
```

`colWeightedVars,dgCMatrix-method`
Calculates the weighted variance for each row (column) of a matrix-like object

---

Description

Calculates the weighted variance for each row (column) of a matrix-like object.

Usage

```
## S4 method for signature 'dgCMatrix'
colWeightedVars(x, w = NULL, rows = NULL, cols = NULL, na.rm = FALSE)

## S4 method for signature 'dgCMatrix'
rowWeightedVars(x, w = NULL, rows = NULL, cols = NULL, na.rm = FALSE)
```

Arguments

`x`	An NxK matrix-like object.
`w`	A `numeric` vector of length K (N) that specifies by how much each element is weighted.
`rows`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`cols`	A `vector` indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
`na.rm`	If `TRUE`, `NA`s are excluded first, otherwise not.

Details

The S4 methods for `x` of type `matrix`, `array`, or `numeric` call `matrixStats::rowWeightedVars` / `matrixStats::colWeightedVars`.

Value

Returns a `numeric` `vector` of length N (K).

See Also

- `matrixStats::rowWeightedVars()` and `matrixStats::colWeightedVars()` which are used when the input is a `matrix` or `numeric` vector.
- See also rowVars for the corresponding unweighted function.

Examples

```
mat <- matrix(rnorm(15), nrow = 5, ncol = 3)
  mat[2, 1] <- NA
  mat[3, 3] <- Inf
  mat[4, 1] <- 0

  print(mat)
  w <- rnorm(n = 5, mean = 3)
  rowWeightedVars(mat, w = w[1:3])
  colWeightedVars(mat, w = w)
```

Index