Package ‘sparseMatrixStats’

May 30, 2024

Type Package
Title Summary Statistics for Rows and Columns of Sparse Matrices
Version 1.16.0
Description High performance functions for row and column operations on sparse matrices. For example: \( \text{col / rowMeans2, col / rowMedians, col / rowVars} \) etc. Currently, the optimizations are limited to data in the column sparse format.
This package is inspired by the \texttt{matrixStats} package by Henrik Bengtsson.

License MIT + file LICENSE
Encoding UTF-8
 LazyData true
RoxygenNote 7.3.1

LinkingTo Rcpp

Imports Rcpp, Matrix, \texttt{matrixStats} (>= 0.60.0), methods
Depends MatrixGenerics (>= 1.5.3)
Suggests testthat (>= 2.1.0), knitr, bench, rmarkdown, BiocStyle

SystemRequirements C++11

VignetteBuilder knitr

URL https://github.com/const-ae/sparseMatrixStats

BugReports https://github.com/const-ae/sparseMatrixStats/issues

biocViews Infrastructure, Software, DataRepresentation

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

git_branch RELEASE_3_19

git_last_commit 2ad650c

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-29

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

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

<table>
<thead>
<tr>
<th>Function</th>
<th>Method</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>colAlls</td>
<td>xgCMatrix-method</td>
<td>3</td>
</tr>
<tr>
<td>colAnyNAs</td>
<td>xgCMatrix-method</td>
<td>4</td>
</tr>
<tr>
<td>colAnyNs</td>
<td>xgCMatrix-method</td>
<td>5</td>
</tr>
<tr>
<td>colAvgsPerRowSet</td>
<td>xgCMatrix-method</td>
<td>7</td>
</tr>
<tr>
<td>colCollapse</td>
<td>xgCMatrix-method</td>
<td>8</td>
</tr>
<tr>
<td>colCounts</td>
<td>xgCMatrix-method</td>
<td>9</td>
</tr>
<tr>
<td>colCummaxs</td>
<td>dgCMatrix-method</td>
<td>11</td>
</tr>
<tr>
<td>colCummins</td>
<td>dgCMatrix-method</td>
<td>12</td>
</tr>
<tr>
<td>colCumprods</td>
<td>xgCMatrix-method</td>
<td>13</td>
</tr>
<tr>
<td>colCumsums</td>
<td>xgCMatrix-method</td>
<td>14</td>
</tr>
<tr>
<td>colDiffs</td>
<td>dgCMatrix-method</td>
<td>15</td>
</tr>
<tr>
<td>colIQRDiff</td>
<td>dgCMatrix-method</td>
<td>17</td>
</tr>
<tr>
<td>colIQRs</td>
<td>xgCMatrix-method</td>
<td>18</td>
</tr>
<tr>
<td>colLogSumExp</td>
<td>xgCMatrix-method</td>
<td>20</td>
</tr>
<tr>
<td>colMadDiffs</td>
<td>dgCMatrix-method</td>
<td>21</td>
</tr>
<tr>
<td>colMads</td>
<td>dgCMatrix-method</td>
<td>23</td>
</tr>
<tr>
<td>colMaxs</td>
<td>dgCMatrix-method</td>
<td>24</td>
</tr>
<tr>
<td>colMeans2</td>
<td>xgCMatrix-method</td>
<td>25</td>
</tr>
<tr>
<td>colMedians</td>
<td>dgCMatrix-method</td>
<td>27</td>
</tr>
<tr>
<td>colMins</td>
<td>dgCMatrix-method</td>
<td>28</td>
</tr>
<tr>
<td>colOrderStats</td>
<td>dgCMatrix-method</td>
<td>29</td>
</tr>
<tr>
<td>colProds</td>
<td>xgCMatrix-method</td>
<td>30</td>
</tr>
<tr>
<td>colQuantiles</td>
<td>xgCMatrix-method</td>
<td>31</td>
</tr>
<tr>
<td>colRanges</td>
<td>dgCMatrix-method</td>
<td>33</td>
</tr>
<tr>
<td>colRanks</td>
<td>dgCMatrix-method</td>
<td>34</td>
</tr>
<tr>
<td>colSdDiffs</td>
<td>dgCMatrix-method</td>
<td>36</td>
</tr>
<tr>
<td>colSds</td>
<td>xgCMatrix-method</td>
<td>38</td>
</tr>
<tr>
<td>colSums2</td>
<td>xgCMatrix-method</td>
<td>39</td>
</tr>
<tr>
<td>colTabulates</td>
<td>xgCMatrix-method</td>
<td>40</td>
</tr>
<tr>
<td>colVarDiffs</td>
<td>dgCMatrix-method</td>
<td>41</td>
</tr>
<tr>
<td>colVars</td>
<td>xgCMatrix-method</td>
<td>43</td>
</tr>
<tr>
<td>colWeightedMads</td>
<td>dgCMatrix-method</td>
<td>44</td>
</tr>
<tr>
<td>colWeightedMeans</td>
<td>xgCMatrix-method</td>
<td>46</td>
</tr>
<tr>
<td>colWeightedMedians</td>
<td>dgCMatrix-method</td>
<td>48</td>
</tr>
<tr>
<td>colWeightedSds</td>
<td>xgCMatrix-method</td>
<td>49</td>
</tr>
<tr>
<td>colWeightedVars</td>
<td>xgCMatrix-method</td>
<td>51</td>
</tr>
<tr>
<td>xgCMatrix-class</td>
<td></td>
<td>52</td>
</tr>
</tbody>
</table>

### Index

53.
**Description**

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

**Usage**

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

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

**Arguments**

- **x**: An NxK matrix-like object.
- **rows, 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, missing values (NA or NaN) are omitted from the calculations.
- **useNames**: If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

**Details**

The S4 methods for x of type matrix, array, table, 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 or numeric vector.
- For checks if any element is equal to a value, see rowAnys().
- base::all().

Examples

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

print(mat)

dRowAlls(mat)
dColAlls(mat)
```

Description

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

Usage

```r
## S4 method for signature 'xgCMatrix'
colAnyNAs(x, rows = NULL, cols = NULL, useNames = TRUE)

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

Arguments

- `x`: An NxK matrix-like object.
- `rows, cols`: A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
- `useNames`: If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.
Details

The S4 methods for \texttt{x} of type \texttt{matrix}, \texttt{array}, \texttt{table}, or \texttt{numeric} call \texttt{matrixStats::rowAnyNAs} / \texttt{matrixStats::colAnyNAs}.

Value

Returns a \texttt{logical vector} of length \(N(K)\).

See Also

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

Examples

```r
mat <- matrix(0, nrow=10, ncol=5)
mat[sample(seq_len(50), 5)] <- NA
sp_mat <- as(mat, "dgCMatrix")
colAnyNas(sp_mat)
```

---

**Description**

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

**Usage**

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

```r
## S4 method for signature 'xgCMatrix'
rowAnys(
  x,
```
Arguments

- **x**: An N\(x\)K matrix-like object.
- **rows, 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`, missing values (NA or NaN) are omitted from the calculations.
- **useNames**: If `TRUE` (default), names attributes of result are set. Else if `FALSE`, no naming support is done.

Details

The S4 methods for `x` of type `matrix`, `array`, `table`, 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.
- For checks if all elements are equal to a value, see `rowAlls()`.
- `base::any()`.

Examples

```r
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)
```
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**

```r
## S4 method for signature 'xgCMatrix'
colAvgsPerRowSet(
  X,
  W = NULL,
  cols = NULL,
  S,
  FUN = colMeans2,
  ...,
  na.rm = NA,
  tFUN = FALSE
)
```

```r
## S4 method for signature 'xgCMatrix'
rowAvgsPerColSet(
  X,
  W = NULL,
  rows = NULL,
  S,
  FUN = rowMeans2,
  ...,
  na.rm = NA,
  tFUN = FALSE
)
```

**Arguments**

- **X**
  - An N×M matrix-like object.

- **W**
  - An optional numeric N×M matrix of weights.

- **S**
  - An integer K×J 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.

- **na.rm**
  - (logical) Argument passed to FUN() as na.rm = na.rm. If NA (default), then na.rm = TRUE is used if X or S holds missing values, otherwise na.rm = FALSE.
colCollapse.xgCMatrix-method

### Description

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

### Usage

```r
## S4 method for signature 'xgCMatrix'
colCollapse(x, idxs, cols = NULL, useNames = TRUE)
## S4 method for signature 'xgCMatrix'
rowCollapse(x, idxs, rows = NULL, useNames = TRUE)
```
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).
- **useNames**: If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.
- **rows, 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, table, 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

```r
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)
```

Description

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

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

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

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

Arguments

- **x**: An NxK matrix-like object.
- **rows, 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, missing values (NA or NaN) are omitted from the calculations.
- **useNames**: If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

Details

The S4 methods for x of type matrix, array, table, 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

```r
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)
```

Description

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

Usage

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

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

Arguments

- `x`: An NxK matrix-like object.
- `rows, cols`: A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
- `useNames`: If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

Details

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

Value

Returns a numeric matrix with the same dimensions as `x`. 
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

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

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

Arguments

- `x`: An NxK matrix-like object.
- `rows, cols`: A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
- `useNames`: If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

Details

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

See Also

- `matrixStats::rowCummaxs()` and `matrixStats::colCummaxs()` which are used when the input is a matrix or numeric vector.
- `rowMaxs()` for single maximum estimates.
- `base::cummax()` for cumulative maxima.

Examples

```r
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)
```

```r
```
Value

Returns a numeric matrix with 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

```r
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, xgCMatrix-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

```r
## S4 method for signature 'xgCMatrix'
colCumprods(x, rows = NULL, cols = NULL, useNames = TRUE)

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

Arguments

- **x**
  - An NxK matrix-like object.
- **rows, cols**
  - A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
- **useNames**
  - If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.
Details
The S4 methods for `x` of type `matrix`, `array`, `table`, or `numeric` call `matrixStats::rowCumprods` / `matrixStats::colCumprods`.

Value
Returns a numeric matrix with 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
```r
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, xgCMatrix-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 'xgCMatrix'
colCumsums(x, rows = NULL, cols = NULL, useNames = TRUE)

## S4 method for signature 'xgCMatrix'
rowCumsums(x, rows = NULL, cols = NULL, useNames = TRUE)
```
Arguments

- **x**: An NxK matrix-like object.
- **rows, cols**: A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
- **useNames**: If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

Details

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

Value

Returns a numeric matrix with 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

```r
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)
```

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,
  useNames = TRUE
)

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

Arguments

x An NxK matrix-like object.
rows, 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.
useNames If TRUE (default), names attributes of result are set. Else if FALSE, no naming
           support is done.

Details

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

Value

Returns a numeric matrix with one column (row) less than x: \( N x (K - 1) \) 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

```r
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)
```

---

**Description**

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

**Usage**

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

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

- **x**: An NxK matrix-like object.
- **rows, cols**: A vector indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
- **na.rm**: If `TRUE`, missing values (NA or NaN) are omitted from the calculations.
- **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.
- **useNames**: If `TRUE` (default), names attributes of result are set. Else if `FALSE`, no naming support is done.

Details

The S4 methods for x of type `matrix`, `array`, `table`, 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

```r
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)
```

Description

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

## S4 method for signature 'xgCMatrix'
colIQRs(x, rows = NULL, cols = NULL, na.rm = FALSE, useNames = TRUE)

## S4 method for signature 'xgCMatrix'
rowIQRs(x, rows = NULL, cols = NULL, na.rm = FALSE, useNames = TRUE)

Arguments

x
An NxK matrix-like object.

rows, cols
A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.

na.rm
If TRUE, missing values (NA or NaN) are omitted from the calculations.

useNames
If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

Details

The S4 methods for x of type matrix, array, table, 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, xgCMatrix-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

```r
## S4 method for signature 'xgCMatrix'
colLogSumExps(lx, rows = NULL, cols = NULL, na.rm = FALSE, useNames = TRUE)
## S4 method for signature 'xgCMatrix'
rowLogSumExps(lx, rows = NULL, cols = NULL, na.rm = FALSE, useNames = TRUE)
```

Arguments

- `lx` An NxK matrix-like object. Typically `lx` are `log(x)` values.
- `rows, cols` A vector indicating the subset (and/or columns) to operate over. If `NULL`, no subsetting is done.
- `na.rm` If `TRUE`, missing values (`NA` or `NaN`) are omitted from the calculations.
- `useNames` If `TRUE` (default), names attributes of result are set. Else if `FALSE`, no naming support is done.

Details

The S4 methods for `x` of type `matrix, array, table, 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

```r
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)
```

Description

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

Usage

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

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

x  An NxK matrix-like object.

rows, cols A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.

na.rm If TRUE, missing values (NA or NaN) are omitted from the calculations.

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.

... Additional arguments passed to specific methods.

useNames If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

Details

The S4 methods for x of type matrix, array, table, 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)
Description

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

Usage

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

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

Arguments

- `x`: An N×K matrix-like object.
- `rows, cols`: A vector indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
- `center`: (optional) the center, defaults to the row means
- `constant`: A scale factor. See `stats::mad()` for details.
- `na.rm`: If `TRUE`, missing values (NA or NaN) are omitted from the calculations.
- `useNames`: If `TRUE` (default), names attributes of result are set. Else if `FALSE`, no naming support is done.
Details

The S4 methods for x of type matrix, array, table, 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

```r
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)
```

Description

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

Usage

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

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

- **x**: An NxK matrix-like object.
- **rows, cols**: A vector indicating the subset of rows (and/or columns) to operate over. If **NULL**, no subsetting is done.
- **na.rm**: If **TRUE**, missing values (NA or NaN) are omitted from the calculations.
- **useNames**: If **TRUE** (default), names attributes of result are set. Else if **FALSE**, no naming support is done.

Details

The S4 methods for **x** of type **matrix**, **array**, **table**, 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

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

## S4 method for signature 'xgCMatrix'
rowMeans2(x, rows = NULL, cols = NULL, na.rm = FALSE, useNames = TRUE)

### Arguments

- **x**: An NxK matrix-like object.
- **rows, cols**: A vector indicating the subset of rows (and/or columns) to operate over. If *NULL*, no subsetting is done.
- **na.rm**: If *TRUE*, missing values (*NA* or *NaN*) are omitted from the calculations.
- **useNames**: If *TRUE* (default), names attributes of result are set. Else if *FALSE*, no naming support is done.

### Details

The S4 methods for `x` of type *matrix, array, table, 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

```r
colMeans2(xgCMatrix)
```

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

Arguments
x
   An NxK matrix-like object.
rows, cols
   A vector indicating the subset of rows (and/or columns) to operate over. If
   NULL, no subsetting is done.
na.rm
   If TRUE, missing (NA or NaN) are omitted from the calculations.
useNames
   If TRUE (default), names attributes of result are set. Else if FALSE, no naming
   support is done.

Details
The S4 methods for x of type matrix, array, table, 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,dgCMatrtx-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**

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

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

**Arguments**

- **x**: An NxK matrix-like object.
- **rows, cols**: A vector indicating the subset of rows (and/or columns) to operate over. If `NULL`, no subsetting is done.
- **na.rm**: If `TRUE`, missing values (`NA` or `NaN`) are omitted from the calculations.
- **useNames**: If `TRUE` (default), names attributes of result are set. Else if `FALSE`, no naming support is done.

**Details**

The S4 methods for `x` of type `matrix`, `array`, `table`, 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)

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, 
useNames = TRUE
)

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

Arguments

x An NxK matrix-like object.

rows, 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.

useNames  If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

Details

The S4 methods for \(x\) of type \texttt{matrix}, \texttt{array}, \texttt{table}, or \texttt{numeric} call \texttt{matrixStats::rowOrderStats} \texttt{/ matrixStats::colOrderStats}.

Value

Returns a \texttt{numeric vector} of length \(N\) \((K)\).

See Also

• \texttt{matrixStats::rowOrderStats()} and \texttt{matrixStats::colOrderStats()} which are used when the input is a matrix or numeric vector.

Examples

```r
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)
```

---

**Description**

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

**Usage**

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

```r
## S4 method for signature 'xgCMatrix'
rowProds(x, rows = NULL, cols = NULL, na.rm = FALSE, useNames = TRUE)
```
Arguments

- **x**: An NxK matrix-like object.
- **rows, cols**: A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
- **na.rm**: If TRUE, missing values (NA or NaN) are omitted from the calculations.
- **useNames**: If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

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

```r
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)
```

Description

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

```r
## S4 method for signature 'xgCMatrix'
colQuantiles(
  x,
  rows = NULL,
  cols = NULL,
  probs = seq(from = 0, to = 1, by = 0.25),
  na.rm = FALSE,
  type = 7L,
  useNames = TRUE,
  drop = TRUE
)
```

```r
## S4 method for signature 'xgCMatrix'
rowQuantiles(
  x,
  rows = NULL,
  cols = NULL,
  probs = seq(from = 0, to = 1, by = 0.25),
  na.rm = FALSE,
  type = 7L,
  useNames = TRUE,
  drop = TRUE
)
```

Arguments

- **x**: An NxK matrix-like object.
- **rows, 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`, missing values (NA or NaN) are omitted from the calculations.
- **type**: An integer specifying the type of estimator. See `stats::quantile()`, for more details.
- **useNames**: If `TRUE` (default), names attributes of result are set. Else if `FALSE`, no naming support is done.
- **drop**: If `TRUE` a vector is returned if J == 1.

Details

The S4 methods for x of type `matrix, array, table, 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

```r
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)
```

---

**Description**

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

**Usage**

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

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

**Arguments**

- `x` An NxK matrix-like object.
- `rows, cols` A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
- `na.rm` If TRUE, missing values (NA or NaN) are omitted from the calculations.
- `useNames` If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

**Details**

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

A numeric \( N \times 2 \) (\( K \times 2 \)) matrix, where \( N \) (\( K \)) is the number of rows (columns) for which the ranges are calculated.

Note

Unfortunately for the argument list of the `rowRanges()` generic function we cannot follow the scheme used for the other row/column matrix summarization generic functions. This is because we need to be compatible with the historic `rowRanges()` getter for `RangedSummarizedExperiment` objects. See ?SummarizedExperiment::rowRanges.

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

```r
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)
```

Description

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

Usage

```r
## S4 method for signature 'dgCMatrix'
colRanks(
  x,
  rows = NULL,
  cols = NULL,
```
### S4 method for signature 'dgCMatrix'

```r
colRanks = c("max", "average", "min"),
preserveShape = FALSE,
na.handling = c("keep", "last"),
...,
useNames = TRUE
)
```

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

**Arguments**

- **x** An NxK matrix-like object.
- **rows, 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.
- **preserveShape** 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
- **...** Additional arguments passed to specific methods.
- **useNames** If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

**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". It has dimensions' NxJ (KxJ) matrix, where N (K) is the number of rows (columns) of the input x.
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

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

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

See Also

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

Examples

```r
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)
```
Arguments

- **x**: An NxK matrix-like object.
- **rows, cols**: A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
- **na.rm**: If TRUE, missing values (NA or NaN) are omitted from the calculations.
- **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.
- **useNames**: If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

Details

The S4 methods for x of type matrix, array, table, 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

```r
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, xgCMatrix-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 'xgCMatrix'
colSds(
  x,
  rows = NULL,
  cols = NULL,
  na.rm = FALSE,
  center = NULL,
  useNames = TRUE
)

## S4 method for signature 'xgCMatrix'
rowSds(
  x,
  rows = NULL,
  cols = NULL,
  na.rm = FALSE,
  center = NULL,
  useNames = TRUE
)

Arguments

x
An NxK matrix-like object.

rows, cols
A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.

na.rm
If TRUE, missing values (NA or NaN) are omitted from the calculations.

center
(optional) the center, defaults to the row means

useNames
If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

Details

The S4 methods for x of type matrix, array, table, 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

```r
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.xgCMatrix-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

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

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

Arguments

- **x**: An NxK matrix-like object.
- **rows, cols**: A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
- **na.rm**: If TRUE, missing values (NA or NaN) are omitted from the calculations.
- **useNames**: If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.
Details
The S4 methods for x of type matrix, array, table, 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, xgCMatrix-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 'xgCMatrix'
colTabulates(x, rows = NULL, cols = NULL, values = NULL, useNames = TRUE)

## S4 method for signature 'xgCMatrix'
rowTabulates(x, rows = NULL, cols = NULL, values = NULL, useNames = TRUE)
Arguments

- **x**: An NxK matrix-like object.
- **rows, 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.
- **useNames**: If **TRUE** (default), names attributes of result are set. Else if **FALSE**, no naming support is done.

Details

The S4 methods for `x` of type `matrix, array, table, 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

```r
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)
```

Description

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

Usage

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

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

Arguments

- **x**: An NxK matrix-like object.
- **rows, cols**: A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
- **na.rm**: If TRUE, missing values (NA or NaN) are omitted from the calculations.
- **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.
- **useNames**: If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

Details

The S4 methods for x of type matrix, array, table, 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

```r
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)
```

Description

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

Usage

```r
## S4 method for signature 'xgCMatrix'
colVars(
  x,
  rows = NULL,
  cols = NULL,
  na.rm = FALSE,
  center = NULL,
  useNames = TRUE
)

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

Arguments

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

na.rm If TRUE, missing values (NA or NaN) are omitted from the calculations.

center (optional) the center, defaults to the row means.

useNames If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

Details

The S4 methods for x of type matrix, array, table, 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)
Usage

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

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

Arguments

- **x**: An N x K matrix-like object.
- **w**: A numeric vector of length K (N) that specifies by how much each element is weighted.
- **rows, cols**: A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
- **na.rm**: If TRUE, missing values (NA or NaN) are omitted from the calculations.
- **constant**: A scale factor. See stats::mad() for details.
- **center**: Not supported at the moment.
- **useNames**: If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

Details

The S4 methods for x of type matrix, array, table, 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

```r
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, xgCMatrix-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

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

```r
## S4 method for signature 'xgCMatrix'
rowWeightedMeans(
x,
w = NULL,
rows = NULL,
cols = NULL,
```
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, cols  A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
na.rm  If TRUE, missing values (NA or NaN) are omitted from the calculations.
useNames  If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

Details

The S4 methods for x of type matrix, array, table, 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)
Calculates the weighted median for each row (column) of a matrix-like object

Usage

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

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

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, cols` A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
- `na.rm` If TRUE, missing values (NA or NaN) are omitted from the calculations.
- `useNames` If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

Details

The S4 methods for `x` of type `matrix, array, table, 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

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

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

---

**colWeightedSds, xgCMatrix-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

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

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

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, cols A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.

na.rm If TRUE, missing values (NA or NaN) are omitted from the calculations.

useNames If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

Details

The S4 methods for x of type matrix, array, table, 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)
Calculates the weighted variance for each row (column) of a matrix-like object.

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

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

### 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, cols**: A vector indicating the subset of rows (and/or columns) to operate over. If NULL, no subsetting is done.
- **na.rm**: If TRUE, missing values (NA or NaN) are omitted from the calculations.
- **useNames**: If TRUE (default), names attributes of result are set. Else if FALSE, no naming support is done.

### Details

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

Returns a numeric vector of length \( N \times K \).

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

```r
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)
```

---

**xgCMatrix-class**

Union of double and logical column-sparse matrices

**Description**

Union of dgCMatrix and lgCMatrix
Index

colAlls, 3, 4
colAlls,xgCMatrix-method, 3
colAnyNAs, 5
colAnyNAs,xgCMatrix-method, 4
colAnyNs, 6
colAnyNs,xgCMatrix-method, 5
colAvgsPerRowSet, 8
colAvgsPerRowSet
(colAvgsPerRowSet,xgCMatrix-method), 7
colAvgsPerRowSet,xgCMatrix-method, 7
colCollapse, 9
colCollapse,xgCMatrix-method, 8
colCounts, 10
colCounts,xgCMatrix-method, 9
colCummaxs, 11, 12
colCummaxs,dgCMatrix-method, 11
colCummins, 12, 13
colCummins,dgCMatrix-method, 12
colCumprods, 14
colCumprods,xgCMatrix-method, 13
colCumsums, 15
colCumsums,xgCMatrix-method, 14
colDiffs, 16
colDiffs,dgCMatrix-method, 15
colIQRDiffs, 18
colIQRDiffs,dgCMatrix-method, 17
colIQRs, 19
colIQRs,xgCMatrix-method, 18
colLogSumExps, 20
colLogSumExps,xgCMatrix-method, 20
colMadDiffs, 22
colMadDiffs,dgCMatrix-method, 21
colMads, 24
colMads,dgCMatrix-method, 23
colMaxs, 25
colMaxs,dgCMatrix-method, 24
colMeans2, 26
colMeans2,xgCMatrix-method, 25
colMedians, 27
colMedians,dgCMatrix-method, 27
colMins, 28
colMins,dgCMatrix-method, 28
colOrderStats, 30
colOrderStats,dgCMatrix-method, 29
colProds, 31
colProds,xgCMatrix-method, 30
colQuantiles, 32, 33
colQuantiles,xgCMatrix-method, 31
colRanges, 33, 34
colRanges,dgCMatrix-method, 33
colRanks, 36
colRanks,dgCMatrix-method, 34
colSdDiffs, 37
colSdDiffs,dgCMatrix-method, 36
colSds, 38, 39
colSds,xgCMatrix-method, 38
colSums2, 40
colSums2,xgCMatrix-method, 39
colTabulates, 41
colTabulates,xgCMatrix-method, 40
colVarDiffs, 42
colVarDiffs,dgCMatrix-method, 41
colVars, 44
colVars,xgCMatrix-method, 43
colWeightedMads, 45, 46
colWeightedMads,dgCMatrix-method, 44
colWeightedMeans, 47
colWeightedMeans,xgCMatrix-method, 46
colWeightedMedians, 48, 49
colWeightedMedians,dgCMatrix-method,
INDEX

colWeightedSds, 50
colWeightedSds, xgCMatrix-method, 49
colWeightedVars, 51, 52
colWeightedVars, xgCMatrix-method, 51
cummax, 12
cummin, 13
cumprod, 14
cumsum, 15
diff, 16
integer, 7, 10, 35
IQR, 19
is.na, 5
logical, 4–6
mad, 22, 23, 45
matrix, 3, 5, 6, 9–16, 18–20, 22, 24–28, 30–32–35, 37, 38, 40–42, 44, 45, 47, 48, 50, 51
NA, 3, 6, 10, 18–20, 22, 23, 25–28, 31–33, 37–39, 42, 44, 45, 47, 48, 50, 51
NaN, 3, 6, 10, 18–20, 22, 23, 25–28, 31–33, 37–39, 42, 44, 45, 47, 48, 50, 51
NULL, 3, 4, 6, 8–13, 15, 16, 18–20, 22, 23, 25–29, 31–33, 35, 37–39, 41, 42, 44, 45, 47, 48, 50, 51
numeric, 3, 5, 6, 9–16, 18–20, 22, 24–28, 30–34, 37–42, 44, 45, 47–52
prod, 31
quantile, 32
range, 34
RangedSummarizedExperiment, 34
rowAlls, 3, 4, 6, 10
rowAlls, xgCMatrix-method
(colAlls, xgCMatrix-method), 3
rowAnyNAs, 5
rowAnyNAs, xgCMatrix-method
(colAnyNAs, xgCMatrix-method), 4
rowAnys, 4–6, 10
rowAnys, xgCMatrix-method
(colAnys, xgCMatrix-method), 5
rowAvgsPerColSet, 8
rowAvgsPerColSet, xgCMatrix-method
(colAvgsPerRowSet, xgCMatrix-method), 7
rowCollapse, 9
rowCollapse, xgCMatrix-method
(colCollapse, xgCMatrix-method), 8
rowCounts, 10
rowCounts, xgCMatrix-method
(colCounts, xgCMatrix-method), 9
rowCummaxs, 11, 12
rowCummaxs, dgCMatrix-method
(colCummaxs, dgCMatrix-method), 11
rowCummins, 12, 13
rowCummins, dgCMatrix-method
(colCummins, dgCMatrix-method), 12
rowCumprods, 14
rowCumprods, xgCMatrix-method
(colCumprods, xgCMatrix-method), 13
rowCumsums, 15
rowCumsums, xgCMatrix-method
(colCumsums, xgCMatrix-method), 14
rowDiffs, 16
rowDiffs, dgCMatrix-method
(colDiffs, dgCMatrix-method), 15
rowIQRDiffs, 18
rowIQRDiffs, dgCMatrix-method
(colIQRDiffs, dgCMatrix-method), 17
rowIQRs, 18, 19
rowIQRs, xgCMatrix-method
(colIQRs, xgCMatrix-method), 18
rowLogSumExps, 20
rowLogSumExps, xgCMatrix-method
(colLogSumExps, xgCMatrix-method), 20
rowMadDiffs, 22
rowMadDiffs, dgCMatrix-method
(colMadDiffs, dgCMatrix-method), 21
rowMads, 24, 46
INDEX

rowMads(), 19
rowMads, dgCMatrix-method
(colMads, dgCMatrix-method), 23
rowMaxs, 12, 23, 28, 34
rowMaxs, dgCMatrix-method
(colMaxs, dgCMatrix-method), 23
rowMeans, 24, 26, 27, 39, 40, 44
rowMeans2, 24, 26, 27, 39, 40, 44, 47
rowMeans2, xgCMatrix-method
(colMeans2, xgCMatrix-method), 25
rowMedians, 27, 49
rowMedians, dgCMatrix-method
(colMedians, dgCMatrix-method), 27
rowMins, 13, 23, 28, 34
rowMins, dgCMatrix-method
(colMins, dgCMatrix-method), 28
rowOrderStats, 30
rowOrderStats, dgCMatrix-method
(colOrderStats, dgCMatrix-method), 29
rowProds, 31
rowProds, xgCMatrix-method
(colProds, xgCMatrix-method), 30
rowQuantiles, 32, 33
rowQuantiles, xgCMatrix-method
(colQuantiles, xgCMatrix-method), 31
rowRanges, 33, 34
rowRanges, dgCMatrix-method
(colRanges, dgCMatrix-method), 33
rowRanks, 36
rowRanks, dgCMatrix-method
(colRanks, dgCMatrix-method), 34
rowVarDiffs, 42
rowVarDiffs, dgCMatrix-method
(colVarDiffs, dgCMatrix-method), 42
rowVars, 26, 39, 44, 52
rowVars(), 43
rowVars, xgCMatrix-method
(colVars, xgCMatrix-method), 43
rowWeightedMads, 45, 46
rowWeightedMads, dgCMatrix-method
(colWeightedMads, dgCMatrix-method), 44
rowWeightedMeans, 47
rowWeightedMeans, xgCMatrix-method
(colWeightedMeans, xgCMatrix-method), 46
rowWeightedMedians, 48, 49
rowWeightedMedians, dgCMatrix-method
(colWeightedMedians, dgCMatrix-method), 48
rowWeightedSds, 50
rowWeightedSds, xgCMatrix-method
(colWeightedSds, xgCMatrix-method), 49
rowWeightedVars, 51, 52
rowWeightedVars, xgCMatrix-method
(colWeightedVars, xgCMatrix-method), 51
stats::quantile, 33
sum, 40
table, 3, 5, 6, 9–12, 14–16, 18–20, 22, 24–28, 30, 32, 33, 37, 38, 40–42, 44, 45, 47, 48, 50, 51
var, 44
vector, 3–6, 8–13, 15, 16, 18–20, 22–33, 35, 37–42, 44, 45, 47–52
xgCMatrix-class, 52