

Package ‘MatrixGenerics’

October 24, 2025

Title S4 Generic Summary Statistic Functions that Operate on Matrix-Like Objects

Description S4 generic functions modeled after the 'matrixStats' API for alternative matrix implementations. Packages with alternative matrix implementation can depend on this package and implement the generic functions that are defined here for a useful set of row and column summary statistics. Other package developers can import this package and handle a different matrix implementations without worrying about incompatibilities.

biocViews Infrastructure, Software

URL <https://bioconductor.org/packages/MatrixGenerics>

BugReports <https://github.com/Bioconductor/MatrixGenerics/issues>

Version 1.21.0

License Artistic-2.0

Encoding UTF-8

Depends matrixStats (>= 1.4.1)

Imports methods

Suggests Matrix, sparseMatrixStats, SparseArray, DelayedArray, DelayedMatrixStats, SummarizedExperiment, testthat (>= 2.1.0)

RoxygenNote 7.3.2

Roxygen list(markdown = TRUE, old_usage = TRUE)

Collate 'MatrixGenerics-package.R' 'rowAlls.R' 'rowAnyNAs.R' 'rowAnys.R' 'rowAvgPerColSet.R' 'rowCollapse.R' 'rowCounts.R' 'rowCummaxs.R' 'rowCummins.R' 'rowCumprods.R' 'rowCumsums.R' 'rowDiffs.R' 'rowIQRDiffs.R' 'rowIQRs.R' 'rowLogSumExps.R' 'rowMadDiffs.R' 'rowMads.R' 'rowMaxs.R' 'rowMeans.R' 'rowMeans2.R' 'rowMedians.R' 'rowMins.R' 'rowOrderStats.R' 'rowProds.R' 'rowQuantiles.R' 'rowRanges.R' 'rowRanks.R' 'rowSdDiffs.R' 'rowSds.R' 'rowSums.R' 'rowSums2.R' 'rowTabulates.R' 'rowVarDiffs.R' 'rowVars.R' 'rowWeightedMads.R' 'rowWeightedMeans.R' 'rowWeightedMedians.R' 'rowWeightedSds.R' 'rowWeightedVars.R'

git_url <https://git.bioconductor.org/packages/MatrixGenerics>

git_branch devel

git_last_commit 2a70895

git_last_commit_date 2025-04-15

Repository Bioconductor 3.23

Date/Publication 2025-10-24

Author Constantin Ahlmann-Eltze [aut] (ORCID:
<https://orcid.org/0000-0002-3762-068X>),
 Peter Hickey [aut, cre] (ORCID:
<https://orcid.org/0000-0002-8153-6258>),
 Hervé Pagès [aut]

Maintainer Peter Hickey <peter.hickey@gmail.com>

Contents

MatrixGenerics-package	3
internal-helpers	3
rowAlls	3
rowAnyNAs	5
rowAnys	6
rowAvgsPerColSet	8
rowCollapse	10
rowCounts	11
rowCummaxs	13
rowCummins	14
rowCumprods	16
rowCumsums	17
rowDiffs	19
rowIQRDiffs	20
rowIQRs	22
rowLogSumExps	23
rowMadDiffs	25
rowMads	26
rowMaxs	28
rowMeans	30
rowMeans2	31
rowMedians	32
rowMins	34
rowOrderStats	35
rowProds	37
rowQuantiles	38
rowRanges	40
rowRanks	42
rowSdDiffs	44
rowSds	45
rowSums	47

rowSums2	48
rowTabulates	49
rowVarDiffs	51
rowVars	53
rowWeightedMads	54
rowWeightedMeans	56
rowWeightedMedians	57
rowWeightedSds	59
rowWeightedVars	60
Index	63

MatrixGenerics-package	
	<i>The MatrixGenerics package</i>

Description

The **MatrixGenerics** package defines S4 generic summary statistic functions that operate on matrix-Like objects.

internal-helpers	<i>Internal helpers</i>
------------------	-------------------------

Description

Not for end users

Usage

normarg_center(center, n, what)

rowAlls	<i>Check if all elements in a row (column) of a matrix-like object are equal to a value</i>
---------	---

Description

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

Usage

```

rowAlls(x, rows = NULL, cols = NULL, value = TRUE, na.rm = FALSE, ...,
        useNames = TRUE)

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

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

colAlls(x, rows = NULL, cols = NULL, value = TRUE, na.rm = FALSE, ...,
        useNames = TRUE)

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

## S4 method for signature 'ANY'
colAlls(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.
...	Additional arguments passed to specific methods.
useNames	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
dim.	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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 `rowAnyis()`.
- `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)
```

rowAnyNAs	<i>Check if any elements in a row (column) of a matrix-like object is missing</i>
-----------	---

Description

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

Usage

```
rowAnyNAs(x, rows = NULL, cols = NULL, ..., useNames = TRUE)

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

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

colAnyNAs(x, rows = NULL, cols = NULL, ..., useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
colAnyNAs(x, rows = NULL,
  cols = NULL, ..., useNames = TRUE)

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

Arguments

<code>x</code>	An NxK matrix-like object.
<code>rows, cols</code>	A vector indicating the subset of rows (and/or columns) to operate over. If NULL , no subsetting is done.
<code>...</code>	Additional arguments passed to specific methods.
<code>useNames</code>	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::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(rnorm(15), nrow = 5, ncol = 3)
mat[2, 1] <- NA
mat[3, 3] <- Inf
mat[4, 1] <- 0

print(mat)

rowAnyNas(mat)
colAnyNas(mat)
```

rowAnys

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

```

rowAnys(x, rows = NULL, cols = NULL, value = TRUE, na.rm = FALSE, ...,
        useNames = TRUE)

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

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

colAnys(x, rows = NULL, cols = NULL, value = TRUE, na.rm = FALSE, ...,
        useNames = TRUE)

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

## S4 method for signature 'ANY'
colAnys(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.
...	Additional arguments passed to specific methods.
useNames	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
dim.	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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

```
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)
```

<code>rowAvsPerColSet</code>	<i>Calculates for each row (column) a summary statistic for equally sized subsets of columns (rows)</i>
------------------------------	---

Description

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

Usage

```
rowAvsPerColSet(X, W = NULL, rows = NULL, S, FUN = rowMeans, ...,
  na.rm = NA, tFUN = FALSE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
rowAvsPerColSet(X, W = NULL,
  rows = NULL, S, FUN = rowMeans, ..., na.rm = NA, tFUN = FALSE)

## S4 method for signature 'ANY'
rowAvsPerColSet(X, W = NULL, rows = NULL, S,
  FUN = rowMeans, ..., na.rm = NA, tFUN = FALSE)

colAvsPerRowSet(X, W = NULL, cols = NULL, S, FUN = colMeans, ...,
  na.rm = NA, tFUN = FALSE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
colAvsPerRowSet(X, W = NULL,
  cols = NULL, S, FUN = colMeans, ..., na.rm = NA, tFUN = FALSE)

## S4 method for signature 'ANY'
```



```
colAvgsPerRowSet(X, W = NULL, cols = NULL, S,
  FUN = colMeans, ..., na.rm = NA, tFUN = FALSE)
```

Arguments

X	An NxM matrix-like object.
W	An optional numeric NxM matrix of weights.
rows, 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.
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.
tFUN	If TRUE, X is transposed before it is passed to FUN.

Details

The S4 methods for x of type [matrix](#), [array](#), [table](#), or [numeric](#) call `matrixStats::rowAvgsPerColSet` / `matrixStats::colAvgsPerRowSet`.

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)
```

rowCollapse

*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

```
rowCollapse(x, idxs, rows = NULL, ..., useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
rowCollapse(x, idxs,
  rows = NULL, dim. = dim(x), ..., useNames = TRUE)

## S4 method for signature 'ANY'
rowCollapse(x, idxs, rows = NULL, ..., useNames = TRUE)

colCollapse(x, idxs, cols = NULL, ..., useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
colCollapse(x, idxs,
  cols = NULL, dim. = dim(x), ..., useNames = TRUE)

## S4 method for signature 'ANY'
colCollapse(x, idxs, cols = 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)
rows, cols	A vector indicating the subset of rows (and/or columns) to operate over. If NULL , no subsetting is done.
...	Additional arguments passed to specific methods.
useNames	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
dim.	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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

```
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)
```

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

Description

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

Usage

```
rowCounts(x, rows = NULL, cols = NULL, value = TRUE, na.rm = FALSE,
  ..., useNames = TRUE)

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

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

colCounts(x, rows = NULL, cols = NULL, value = TRUE, na.rm = FALSE,
  ..., useNames = TRUE)
```

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

## S4 method for signature 'ANY'
colCounts(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.
...	Additional arguments passed to specific methods.
useNames	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
dim.	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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

```
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)

```

rowCummaxs	<i>Calculates the cumulative maxima for each row (column) of a matrix-like object</i>
------------	---

Description

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

Usage

```

rowCummaxs(x, rows = NULL, cols = NULL, ..., useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
rowCummaxs(x, rows = NULL,
  cols = NULL, dim. = dim(x), ..., useNames = TRUE)

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

colCummaxs(x, rows = NULL, cols = NULL, ..., useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
colCummaxs(x, rows = NULL,
  cols = NULL, dim. = dim(x), ..., useNames = TRUE)

## S4 method for signature 'ANY'
colCummaxs(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.
...	Additional arguments passed to specific methods.
useNames	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
dim.	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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`.

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)
```

rowCummins

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

```
rowCummins(x, rows = NULL, cols = NULL, ..., useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
rowCummins(x, rows = NULL,
  cols = NULL, dim. = dim(x), ..., useNames = TRUE)

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

```
colCummins(x, rows = NULL, cols = NULL, ..., useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
colCummins(x, rows = NULL,
  cols = NULL, dim. = dim(x), ..., useNames = TRUE)

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

Arguments

<code>x</code>	An <code>NxK</code> matrix-like object.
<code>rows, cols</code>	A vector indicating the subset of rows (and/or columns) to operate over. If NULL , no subsetting is done.
<code>...</code>	Additional arguments passed to specific methods.
<code>useNames</code>	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
<code>dim.</code>	An integer vector of length two specifying the dimension of <code>x</code> , essential when <code>x</code> is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

Details

The S4 methods for `x` of type [matrix](#), [array](#), [table](#), or [numeric](#) call `matrixStats::rowCummins` / `matrixStats::colCummins`.

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

```
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)
```

rowCumprods	<i>Calculates the cumulative product for each row (column) of a matrix-like object</i>
-------------	--

Description

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

Usage

```
rowCumprods(x, rows = NULL, cols = NULL, ..., useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
rowCumprods(x, rows = NULL,
  cols = NULL, dim. = dim(x), ..., useNames = TRUE)

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

colCumprods(x, rows = NULL, cols = NULL, ..., useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
colCumprods(x, rows = NULL,
  cols = NULL, dim. = dim(x), ..., useNames = TRUE)

## S4 method for signature 'ANY'
colCumprods(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.
...	Additional arguments passed to specific methods.
useNames	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
dim.	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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

```
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)
```

rowCumsums	<i>Calculates the cumulative sum for each row (column) of a matrix-like object</i>
------------	--

Description

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

Usage

```
rowCumsums(x, rows = NULL, cols = NULL, ..., useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
rowCumsums(x, rows = NULL,
  cols = NULL, dim. = dim(x), ..., useNames = TRUE)

## S4 method for signature 'ANY'
rowCumsums(x, rows = NULL, cols = NULL, ...,
  useNames = TRUE)

colCumsums(x, rows = NULL, cols = NULL, ..., useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
colCumsums(x, rows = NULL,
  cols = NULL, dim. = dim(x), ..., useNames = TRUE)
```

```
## S4 method for signature 'ANY'
colCumsums(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.
...	Additional arguments passed to specific methods.
useNames	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
dim.	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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

```
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)
```

rowDiffs	<i>Calculates the difference between each element of a row (column) of a matrix-like object</i>
----------	---

Description

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

Usage

```
rowDiffs(x, rows = NULL, cols = NULL, lag = 1L, differences = 1L, ...,
         useNames = TRUE)
```

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

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

```
colDiffs(x, rows = NULL, cols = NULL, lag = 1L, differences = 1L, ...,
         useNames = TRUE)
```

```
## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
colDiffs(x, rows = NULL,
         cols = NULL, lag = 1L, differences = 1L, dim. = dim(x), ...,
         useNames = TRUE)
```

```
## S4 method for signature 'ANY'
colDiffs(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.
...	Additional arguments passed to specific methods.
useNames	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
dim.	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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`: $Nx(K - 1)$ 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)
```

rowIQRDiffs

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

```
rowIQRDiffs(x, rows = NULL, cols = NULL, na.rm = FALSE, diff = 1L,
  trim = 0, ..., useNames = TRUE)

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

## S4 method for signature 'ANY'
rowIQRDiffs(x, rows = NULL, cols = NULL, na.rm = FALSE,
```

```

diff = 1L, trim = 0, ..., useNames = TRUE)

colIQRDiffs(x, rows = NULL, cols = NULL, na.rm = FALSE, diff = 1L,
  trim = 0, ..., useNames = TRUE)

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

## S4 method for signature 'ANY'
colIQRDiffs(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.
...	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::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)
```

rowIQRs	<i>Calculates the interquartile range for each row (column) of a matrix-like object</i>
---------	---

Description

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

Usage

```
rowIQRs(x, rows = NULL, cols = NULL, na.rm = FALSE, ...,
        useNames = TRUE)

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

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

colIQRs(x, rows = NULL, cols = NULL, na.rm = FALSE, ...,
        useNames = TRUE)

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

## S4 method for signature 'ANY'
colIQRs(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.
...	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::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)
```

rowLogSumExps	<i>Accurately calculates the logarithm of the sum of exponentials for each row (column) of a matrix-like object</i>
---------------	---

Description

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

Usage

```
rowLogSumExps(lx, rows = NULL, cols = NULL, na.rm = FALSE, ...,
  useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
rowLogSumExps(lx,
  rows = NULL, cols = NULL, na.rm = FALSE, dim. = dim(lx), ...,
  useNames = TRUE)

## S4 method for signature 'ANY'
```

```

rowLogSumExps(lx, rows = NULL, cols = NULL,
  na.rm = FALSE, ..., useNames = TRUE)

colLogSumExps(lx, rows = NULL, cols = NULL, na.rm = FALSE, ...,
  useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
colLogSumExps(lx,
  rows = NULL, cols = NULL, na.rm = FALSE, dim. = dim(lx), ...,
  useNames = TRUE)

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

```

Arguments

<code>lx</code>	An NxK matrix-like object. Typically <code>lx</code> are <code>log(x)</code> values.
<code>rows, cols</code>	A vector indicating the subset (and/or columns) to operate over. If <code>NULL</code> , no subsetting is done.
<code>na.rm</code>	If <code>TRUE</code> , missing values (<code>NA</code> or <code>NaN</code>) are omitted from the calculations.
<code>...</code>	Additional arguments passed to specific methods.
<code>useNames</code>	If <code>TRUE</code> (default), names attributes of result are set. Else if <code>FALSE</code> , no naming support is done.
<code>dim.</code>	An integer vector of length two specifying the dimension of <code>x</code> , essential when <code>x</code> is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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

```

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)

```

rowMadDiffs	<i>Calculates the mean absolute deviation of the difference between each element of a row (column) of a matrix-like object</i>
-------------	--

Description

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

Usage

```

rowMadDiffs(x, rows = NULL, cols = NULL, na.rm = FALSE, diff = 1L,
  trim = 0, ..., useNames = TRUE)

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

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

colMadDiffs(x, rows = NULL, cols = NULL, na.rm = FALSE, diff = 1L,
  trim = 0, ..., useNames = TRUE)

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

## S4 method for signature 'ANY'
colMadDiffs(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.

<code>na.rm</code>	If <code>TRUE</code> , missing values (<code>NA</code> or <code>NaN</code>) are omitted from the calculations.
<code>diff</code>	An integer specifying the order of difference.
<code>trim</code>	A double in $[0, 1/2]$ specifying the fraction of observations to be trimmed from each end of (sorted) <code>x</code> before estimation.
<code>...</code>	Additional arguments passed to specific methods.
<code>useNames</code>	If <code>TRUE</code> (default), names attributes of result are set. Else if <code>FALSE</code> , 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)
```

<code>rowMads</code>	<i>Calculates the median absolute deviation for each row (column) of a matrix-like object</i>
----------------------	---

Description

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

Usage

```

rowMads(x, rows = NULL, cols = NULL, center = NULL, constant = 1.4826,
        na.rm = FALSE, ..., useNames = TRUE)

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

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

colMads(x, rows = NULL, cols = NULL, center = NULL, constant = 1.4826,
        na.rm = FALSE, ..., useNames = TRUE)

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

## S4 method for signature 'ANY'
colMads(x, rows = NULL, cols = NULL, center = NULL,
        constant = 1.4826, 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.
center	(optional) the center, defaults to the row means
constant	A scale factor. See <code>stats::mad()</code> for details.
na.rm	If TRUE , missing values (NA or NaN) are omitted from the calculations.
...	Additional arguments passed to specific methods.
useNames	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
dim.	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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

```
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)
```

rowMaxs

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

```
rowMaxs(x, rows = NULL, cols = NULL, na.rm = FALSE, ...,
        useNames = TRUE)

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

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

colMaxs(x, rows = NULL, cols = NULL, na.rm = FALSE, ...,
        useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
colMaxs(x, rows = NULL,
```

```

      cols = NULL, na.rm = FALSE, dim. = dim(x), ..., useNames = TRUE)

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

```

Arguments

<code>x</code>	An <code>NxK</code> matrix-like object.
<code>rows, cols</code>	A vector indicating the subset of rows (and/or columns) to operate over. If <code>NULL</code> , no subsetting is done.
<code>na.rm</code>	If <code>TRUE</code> , missing values (<code>NA</code> or <code>NaN</code>) are omitted from the calculations.
<code>...</code>	Additional arguments passed to specific methods.
<code>useNames</code>	If <code>TRUE</code> (default), names attributes of result are set. Else if <code>FALSE</code> , no naming support is done.
<code>dim.</code>	An integer vector of length two specifying the dimension of <code>x</code> , essential when <code>x</code> is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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

```

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)

```

rowMeans

*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

```
rowMeans(x, na.rm = FALSE, dims = 1, ...)
```

```
colMeans(x, na.rm = FALSE, dims = 1, ...)
```

Arguments

<code>x</code>	An NxK matrix-like object, a numeric data frame, or an array-like object of two or more dimensions.
<code>na.rm</code>	If <code>TRUE</code> , missing values (NA or NaN) are omitted from the calculations.
<code>dims</code>	A single integer indicating which dimensions are regarded as rows or columns to mean over. For <code>rowMeans</code> , the mean is over dimensions <code>dims+1, ...</code> ; for <code>colMeans</code> it is over dimensions <code>1:dims</code> .
<code>...</code>	Additional arguments passed to specific methods.

Details

This man page documents the `rowMeans` and `colMeans` *S4 generic functions* defined in the **Matrix-Generics** package. See `?base::colMeans` for the default methods (defined in the **base** package).

Value

Returns a [numeric vector](#) of length N (K).

See Also

- `base::colMeans` for the default `rowMeans` and `colMeans` methods.
- `Matrix::colMeans` in the **Matrix** package for `rowMeans` and `colMeans` methods defined for `CsparseMatrix` derivatives (e.g. `dgCMatrix` objects).

Examples

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

print(mat)
```

```
rowMeans(mat)
colMeans(mat)
```

rowMeans2

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

```
rowMeans2(x, rows = NULL, cols = NULL, na.rm = FALSE, ...,
          useNames = TRUE)

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

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

colMeans2(x, rows = NULL, cols = NULL, na.rm = FALSE, ...,
          useNames = TRUE)

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

## S4 method for signature 'ANY'
colMeans2(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.
...	Additional arguments passed to specific methods.
useNames	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
dim.	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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

```
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)
```

rowMedians

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

```
rowMedians(x, rows = NULL, cols = NULL, na.rm = FALSE, ...,
  useNames = TRUE)

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

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



```
colMedians(x, rows = NULL, cols = NULL, na.rm = FALSE, ...,
  useNames = TRUE)

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

## S4 method for signature 'ANY'
colMedians(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.
...	Additional arguments passed to specific methods.
useNames	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
dim.	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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)
```

rowMins

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

```
rowMins(x, rows = NULL, cols = NULL, na.rm = FALSE, ...,
        useNames = TRUE)

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

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

colMins(x, rows = NULL, cols = NULL, na.rm = FALSE, ...,
        useNames = TRUE)

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

## S4 method for signature 'ANY'
colMins(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.
...	Additional arguments passed to specific methods.
useNames	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
dim.	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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)
```

rowOrderStats	<i>Calculates an order statistic for each row (column) of a matrix-like object</i>
---------------	--

Description

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

Usage

```
rowOrderStats(x, rows = NULL, cols = NULL, which, ..., useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
rowOrderStats(x, rows = NULL,
  cols = NULL, which, dim. = dim(x), ..., useNames = TRUE)

## S4 method for signature 'ANY'
rowOrderStats(x, rows = NULL, cols = NULL, which, ...,
  useNames = TRUE)

colOrderStats(x, rows = NULL, cols = NULL, which, ..., useNames = TRUE)
```

```
## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
colOrderStats(x, rows = NULL,
  cols = NULL, which, dim. = dim(x), ..., useNames = TRUE)

## S4 method for signature 'ANY'
colOrderStats(x, rows = NULL, cols = NULL, which, ...,
  useNames = TRUE)
```

Arguments

<code>x</code>	An NxK matrix-like object.
<code>rows, cols</code>	A vector indicating the subset of rows (and/or columns) to operate over. If NULL , no subsetting is done.
<code>which</code>	An integer index in [1,K] ([1,N]) indicating which order statistic to be returned
<code>...</code>	Additional arguments passed to specific methods.
<code>useNames</code>	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
<code>dim.</code>	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

Details

The S4 methods for x of type [matrix](#), [array](#), [table](#), 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)
```

rowProds	<i>Calculates the product for each row (column) of a matrix-like object</i>
----------	---

Description

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

Usage

```
rowProds(x, rows = NULL, cols = NULL, na.rm = FALSE, ...,
         useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
rowProds(x, rows = NULL,
         cols = NULL, na.rm = FALSE, method = c("direct", "expSumLog"), ...,
         useNames = TRUE)

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

colProds(x, rows = NULL, cols = NULL, na.rm = FALSE, ...,
         useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
colProds(x, rows = NULL,
         cols = NULL, na.rm = FALSE, method = c("direct", "expSumLog"), ...,
         useNames = TRUE)

## S4 method for signature 'ANY'
colProds(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.
...	Additional arguments passed to specific methods.
useNames	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
method	A character vector of length one that specifies the how the product is calculated. Note, that this is not a generic argument and not all implementation have to provide it.

Details

The S4 methods for `x` of type `matrix`, `array`, `table`, or `numeric` call `matrixStats::rowProds` / `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)
```

rowQuantiles

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

Description

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

Usage

```
rowQuantiles(x, rows = NULL, cols = NULL, probs = seq(from = 0, to = 1,
  by = 0.25), na.rm = FALSE, type = 7L, digits = 7L, ...,
  useNames = TRUE, drop = TRUE)

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

## S4 method for signature 'ANY'
rowQuantiles(x, rows = NULL, cols = NULL,
```

```

probs = seq(from = 0, to = 1, by = 0.25), na.rm = FALSE, type = 7L,
digits = 7L, ..., useNames = TRUE, drop = TRUE)

colQuantiles(x, rows = NULL, cols = NULL, probs = seq(from = 0, to = 1,
  by = 0.25), na.rm = FALSE, type = 7L, digits = 7L, ...,
  useNames = TRUE, drop = TRUE)

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

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

```

Arguments

<code>x</code>	An NxK matrix-like object.
<code>rows, cols</code>	A vector indicating the subset of rows (and/or columns) to operate over. If NULL , no subsetting is done.
<code>probs</code>	A numeric vector of J probabilities in [0, 1].
<code>na.rm</code>	If TRUE , missing values (NA or NaN) are omitted from the calculations.
<code>type</code>	An integer specifying the type of estimator. See <code>stats::quantile()</code> for more details.
<code>digits</code>	An integer specifying the precision of the formatted percentages. See <code>stats::quantile()</code> for more details.
<code>...</code>	Additional arguments passed to specific methods.
<code>useNames</code>	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
<code>drop</code>	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

```

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)

```

rowRanges	<i>Calculates the minimum and maximum for each row (column) of a matrix-like object</i>
-----------	---

Description

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

Usage

```

rowRanges(x, ...)

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

## S4 method for signature 'ANY'
rowRanges(x, ...)

colRanges(x, rows = NULL, cols = NULL, na.rm = FALSE, ...,
  useNames = TRUE)

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

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

```

Arguments

x	An NxK matrix-like object.
...	Additional arguments passed to specific methods.
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.
dim.	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.
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** Nx2 (Kx2) **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

```
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)
```

rowRanks	<i>Calculates the rank of the elements for each row (column) of a matrix-like object</i>
----------	--

Description

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

Usage

```
rowRanks(x, rows = NULL, cols = NULL, ties.method = c("max", "average"),
  ..., useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
rowRanks(x, rows = NULL,
  cols = NULL, ties.method = c("max", "average", "first", "last", "random",
    "max", "min", "dense"), dim. = dim(x), ..., useNames = TRUE)

## S4 method for signature 'ANY'
rowRanks(x, rows = NULL, cols = NULL,
  ties.method = c("max", "average"), ..., useNames = TRUE)

colRanks(x, rows = NULL, cols = NULL, ties.method = c("max", "average"),
  ..., useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
colRanks(x, rows = NULL,
  cols = NULL, ties.method = c("max", "average", "first", "last", "random",
    "max", "min", "dense"), dim. = dim(x), preserveShape = FALSE, ...,
  useNames = TRUE)

## S4 method for signature 'ANY'
colRanks(x, rows = NULL, cols = NULL,
  ties.method = c("max", "average"), ..., 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.
...	Additional arguments passed to specific methods.
useNames	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.

dim.	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.
preserveShape	If TRUE the output matrix has the same shape as the input x. 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**, **table**, or **numeric** call `matrixStats::rowRanks` / `matrixStats::colRanks`.

The `matrixStats::rowRanks()` function can handle a lot of different values for the `ties.method` argument. Users of the generic function should however only rely on `max` and `average` because the other ones are not guaranteed to be implemented:

`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**.

Value

A **matrix** of type **integer** is returned, unless `ties.method = "average"` when it is of type **numeric**.

The `rowRanks()` function always returns an NxK **matrix**, where N (K) is the number of rows (columns) whose ranks are calculated.

The `colRanks()` function returns an NxK **matrix**, if `preserveShape = TRUE`, otherwise a KxN **matrix**.

Any **names** of x are ignored and absent in the result.

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)
```

rowSdDiffs	<i>Calculates the standard deviation of the difference between each element of a row (column) of a matrix-like object</i>
------------	---

Description

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

Usage

```
rowSdDiffs(x, rows = NULL, cols = NULL, na.rm = FALSE, diff = 1L,
           trim = 0, ..., useNames = TRUE)
```

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

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

```
colSdDiffs(x, rows = NULL, cols = NULL, na.rm = FALSE, diff = 1L,
           trim = 0, ..., useNames = TRUE)
```

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

```
## S4 method for signature 'ANY'
colSdDiffs(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.
...	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::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)
```

rowSds	<i>Calculates the standard deviation for each row (column) of a matrix-like object</i>
--------	--

Description

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

Usage

```
rowSds(x, rows = NULL, cols = NULL, na.rm = FALSE, center = NULL, ...,
       useNames = TRUE)

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

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

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

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

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

Arguments

<code>x</code>	An $N \times K$ matrix-like object.
<code>rows, cols</code>	A vector indicating the subset of rows (and/or columns) to operate over. If NULL , no subsetting is done.
<code>na.rm</code>	If TRUE , missing values (NA or NaN) are omitted from the calculations.
<code>center</code>	(optional) the center, defaults to the row means
<code>...</code>	Additional arguments passed to specific methods.
<code>useNames</code>	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
<code>dim.</code>	An integer vector of length two specifying the dimension of <code>x</code> , essential when <code>x</code> is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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

```
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)
```

rowSums

*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

```
rowSums(x, na.rm = FALSE, dims = 1, ...)

colSums(x, na.rm = FALSE, dims = 1, ...)
```

Arguments

<code>x</code>	An NxK matrix-like object, a numeric data frame, or an array-like object of two or more dimensions.
<code>na.rm</code>	If TRUE , missing values (NA or NaN) are omitted from the calculations.
<code>dims</code>	A single integer indicating which dimensions are regarded as rows or columns to sum over. For <code>rowSums</code> , the sum is over dimensions <code>dims+1, ...</code> ; for <code>colSums</code> it is over dimensions <code>1:dims</code> .
<code>...</code>	Additional arguments passed to specific methods.

Details

This man page documents the `rowSums` and `colSums` *S4 generic functions* defined in the **Matrix-Generics** package. See `?base::colSums` for the default methods (defined in the **base** package).

Value

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

See Also

- `base::colSums` for the default `rowSums` and `colSums` methods.
- `Matrix::colSums` in the **Matrix** package for `rowSums` and `colSums` methods defined for `CsparseMatrix` derivatives (e.g. `dgCMatrix` objects).

Examples

```

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

print(mat)

rowSums(mat)
colSums(mat)

```

rowSums2

*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

```

rowSums2(x, rows = NULL, cols = NULL, na.rm = FALSE, ...,
         useNames = TRUE)

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

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

colSums2(x, rows = NULL, cols = NULL, na.rm = FALSE, ...,
         useNames = TRUE)

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

## S4 method for signature 'ANY'
colSums2(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.
...	Additional arguments passed to specific methods.
useNames	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
dim.	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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)
```

rowTabulates

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

Description

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

Usage

```

rowTabulates(x, rows = NULL, cols = NULL, values = NULL, ...,
  useNames = TRUE)

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

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

colTabulates(x, rows = NULL, cols = NULL, values = NULL, ...,
  useNames = TRUE)

## S4 method for signature 'matrix_OR_array_OR_table_OR_numeric'
colTabulates(x, rows = NULL,
  cols = NULL, values = NULL, ..., useNames = TRUE)

## S4 method for signature 'ANY'
colTabulates(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.
...	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::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)

```

rowVarDiffs	<i>Calculates the variance of the difference between each element of a row (column) of a matrix-like object</i>
-------------	---

Description

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

Usage

```

rowVarDiffs(x, rows = NULL, cols = NULL, na.rm = FALSE, diff = 1L,
  trim = 0, ..., useNames = TRUE)

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

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

colVarDiffs(x, rows = NULL, cols = NULL, na.rm = FALSE, diff = 1L,
  trim = 0, ..., useNames = TRUE)

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

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

```

Arguments

<code>x</code>	An NxK matrix-like object.
<code>rows, cols</code>	A vector indicating the subset of rows (and/or columns) to operate over. If NULL , no subsetting is done.
<code>na.rm</code>	If TRUE , missing values (NA or NaN) are omitted from the calculations.
<code>diff</code>	An integer specifying the order of difference.
<code>trim</code>	A double in [0,1/2] specifying the fraction of observations to be trimmed from each end of (sorted) <code>x</code> before estimation.
<code>...</code>	Additional arguments passed to specific methods.
<code>useNames</code>	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

```
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)
```

rowVars	<i>Calculates the variance for each row (column) of a matrix-like object</i>
---------	--

Description

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

Usage

```
rowVars(x, rows = NULL, cols = NULL, na.rm = FALSE, center = NULL, ...,
        useNames = TRUE)
```

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

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

```
colVars(x, rows = NULL, cols = NULL, na.rm = FALSE, center = NULL, ...,
        useNames = TRUE)
```

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

```
## S4 method for signature 'ANY'
colVars(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.
...	Additional arguments passed to specific methods.
useNames	If TRUE (default), names attributes of result are set. Else if FALSE , no naming support is done.
dim.	An integer vector of length two specifying the dimension of x, essential when x is a numeric vector. Note, that this is not a generic argument and not all methods need provide it.

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)
```

rowWeightedMads	<i>Calculates the weighted median absolute deviation for each row (column) of a matrix-like object</i>
-----------------	--

Description

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

Usage

```
rowWeightedMads(x, w = NULL, rows = NULL, cols = NULL, na.rm = FALSE,
  constant = 1.4826, center = NULL, ..., useNames = TRUE)

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

## S4 method for signature 'ANY'
```

```

rowWeightedMads(x, w = NULL, rows = NULL, cols = NULL,
  na.rm = FALSE, constant = 1.4826, center = NULL, ...,
  useNames = TRUE)

colWeightedMads(x, w = NULL, rows = NULL, cols = NULL, na.rm = FALSE,
  constant = 1.4826, center = NULL, ..., useNames = TRUE)

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

## S4 method for signature 'ANY'
colWeightedMads(x, w = NULL, rows = NULL, cols = NULL,
  na.rm = FALSE, constant = 1.4826, center = NULL, ...,
  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.
constant	A scale factor. See <code>stats::mad()</code> for details.
center	(optional) the center, defaults to the row means
...	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::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(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)
rowWeightedMads(mat, w = w[1:3])
colWeightedMads(mat, w = w)

```

rowWeightedMeans	<i>Calculates the weighted mean for each row (column) of a matrix-like object</i>
------------------	---

Description

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

Usage

```

rowWeightedMeans(x, w = NULL, rows = NULL, cols = NULL, na.rm = FALSE,
  ..., useNames = TRUE)

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

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

colWeightedMeans(x, w = NULL, rows = NULL, cols = NULL, na.rm = FALSE,
  ..., useNames = TRUE)

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

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

```

Arguments

x An NxK matrix-like object.

<code>w</code>	A numeric vector of length K (N) that specifies by how much each element is weighted.
<code>rows, cols</code>	A vector indicating the subset of rows (and/or columns) to operate over. If NULL , no subsetting is done.
<code>na.rm</code>	If TRUE , missing values (NA or NaN) are omitted from the calculations.
<code>...</code>	Additional arguments passed to specific methods.
<code>useNames</code>	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)
```

<code>rowWeightedMedians</code>	<i>Calculates the weighted median for each row (column) of a matrix-like object</i>
---------------------------------	---

Description

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

Usage

```

rowWeightedMedians(x, w = NULL, rows = NULL, cols = NULL,
  na.rm = FALSE, ..., useNames = TRUE)

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

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

colWeightedMedians(x, w = NULL, rows = NULL, cols = NULL,
  na.rm = FALSE, ..., useNames = TRUE)

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

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

```

Arguments

<code>x</code>	An $N \times K$ matrix-like object.
<code>w</code>	A numeric vector of length K (N) that specifies by how much each element is weighted.
<code>rows, cols</code>	A vector indicating the subset of rows (and/or columns) to operate over. If NULL , no subsetting is done.
<code>na.rm</code>	If TRUE , missing values (NA or NaN) are omitted from the calculations.
<code>...</code>	Additional arguments passed to specific methods.
<code>useNames</code>	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

```
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)
```

rowWeightedSds	<i>Calculates the weighted standard deviation for each row (column) of a matrix-like object</i>
----------------	---

Description

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

Usage

```
rowWeightedSds(x, w = NULL, rows = NULL, cols = NULL, na.rm = FALSE,
  ..., useNames = TRUE)

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

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

colWeightedSds(x, w = NULL, rows = NULL, cols = NULL, na.rm = FALSE,
  ..., useNames = TRUE)

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

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

Arguments

<code>x</code>	An NxK matrix-like object.
<code>w</code>	A numeric vector of length K (N) that specifies by how much each element is weighted.
<code>rows, cols</code>	A vector indicating the subset of rows (and/or columns) to operate over. If NULL , no subsetting is done.
<code>na.rm</code>	If TRUE , missing values (NA or NaN) are omitted from the calculations.
<code>...</code>	Additional arguments passed to specific methods.
<code>useNames</code>	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)
```

<code>rowWeightedVars</code>	<i>Calculates the weighted variance for each row (column) of a matrix-like object</i>
------------------------------	---

Description

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

Usage

```

rowWeightedVars(x, w = NULL, rows = NULL, cols = NULL, na.rm = FALSE,
  ..., useNames = TRUE)

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

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

colWeightedVars(x, w = NULL, rows = NULL, cols = NULL, na.rm = FALSE,
  ..., useNames = TRUE)

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

## S4 method for signature 'ANY'
colWeightedVars(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.
...	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::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

- * **algebra**
 - rowMeans, [30](#)
 - rowSums, [47](#)
- * **arith**
 - rowMeans, [30](#)
 - rowSums, [47](#)
- * **array**
 - rowAvgPerColSet, [8](#)
 - rowCollapse, [10](#)
 - rowCummaxs, [13](#)
 - rowCummins, [14](#)
 - rowCumprods, [16](#)
 - rowCumsums, [17](#)
 - rowDiffs, [19](#)
 - rowIQRDiffs, [20](#)
 - rowIQRs, [22](#)
 - rowLogSumExps, [23](#)
 - rowMadDiffs, [25](#)
 - rowMads, [26](#)
 - rowMaxs, [28](#)
 - rowMeans, [30](#)
 - rowMeans2, [31](#)
 - rowMedians, [32](#)
 - rowMins, [34](#)
 - rowOrderStats, [35](#)
 - rowProds, [37](#)
 - rowQuantiles, [38](#)
 - rowRanges, [40](#)
 - rowRanks, [42](#)
 - rowSdDiffs, [44](#)
 - rowSds, [45](#)
 - rowSums, [47](#)
 - rowSums2, [48](#)
 - rowTabulates, [49](#)
 - rowVarDiffs, [51](#)
 - rowVars, [53](#)
 - rowWeightedMads, [54](#)
 - rowWeightedMeans, [56](#)
 - rowWeightedMedians, [57](#)
 - rowWeightedSds, [59](#)
 - rowWeightedVars, [60](#)
- * **internal**
 - internal-helpers, [3](#)
- * **iteration**
 - rowAvgPerColSet, [8](#)
 - rowCollapse, [10](#)
 - rowCummaxs, [13](#)
 - rowCummins, [14](#)
 - rowCumprods, [16](#)
 - rowCumsums, [17](#)
 - rowDiffs, [19](#)
 - rowIQRDiffs, [20](#)
 - rowIQRs, [22](#)
 - rowLogSumExps, [23](#)
 - rowMadDiffs, [25](#)
 - rowMads, [26](#)
 - rowMaxs, [28](#)
 - rowMeans, [30](#)
 - rowMeans2, [31](#)
 - rowMedians, [32](#)
 - rowMins, [34](#)
 - rowOrderStats, [35](#)
 - rowProds, [37](#)
 - rowQuantiles, [38](#)
 - rowRanges, [40](#)
 - rowRanks, [42](#)
 - rowSdDiffs, [44](#)
 - rowSds, [45](#)
 - rowSums, [47](#)
 - rowSums2, [48](#)
 - rowTabulates, [49](#)
 - rowVarDiffs, [51](#)
 - rowVars, [53](#)
 - rowWeightedMads, [54](#)
 - rowWeightedMeans, [56](#)
 - rowWeightedMedians, [57](#)
 - rowWeightedSds, [59](#)
 - rowWeightedVars, [60](#)

*** robust**

- rowAvgsPerColSet, 8
- rowCollapse, 10
- rowCummaxs, 13
- rowCummins, 14
- rowCumprods, 16
- rowCumsums, 17
- rowDiffs, 19
- rowIQRDiffs, 20
- rowIQRs, 22
- rowLogSumExps, 23
- rowMadDiffs, 25
- rowMads, 26
- rowMaxs, 28
- rowMeans, 30
- rowMeans2, 31
- rowMedians, 32
- rowMins, 34
- rowOrderStats, 35
- rowProds, 37
- rowQuantiles, 38
- rowRanges, 40
- rowRanks, 42
- rowSdDiffs, 44
- rowSds, 45
- rowSums, 47
- rowSums2, 48
- rowTabulates, 49
- rowVarDiffs, 51
- rowVars, 53
- rowWeightedMads, 54
- rowWeightedMeans, 56
- rowWeightedMedians, 57
- rowWeightedSds, 59
- rowWeightedVars, 60

*** univar2**

- rowAvgsPerColSet, 8

*** univar**

- rowCollapse, 10
- rowCummaxs, 13
- rowCummins, 14
- rowCumprods, 16
- rowCumsums, 17
- rowDiffs, 19
- rowIQRDiffs, 20
- rowIQRs, 22
- rowLogSumExps, 23
- rowMadDiffs, 25

- rowMads, 26
- rowMaxs, 28
- rowMeans, 30
- rowMeans2, 31
- rowMedians, 32
- rowMins, 34
- rowOrderStats, 35
- rowProds, 37
- rowSdDiffs, 44
- rowSds, 45
- rowSums, 47
- rowSums2, 48
- rowTabulates, 49
- rowVarDiffs, 51
- rowVars, 53
- rowWeightedMads, 54
- rowWeightedMeans, 56
- rowWeightedMedians, 57
- rowWeightedSds, 59
- rowWeightedVars, 60

*** utilities**

- internal-helpers, 3

all, 5

any, 6, 8

array, 4, 6, 7, 9, 10, 12, 14–16, 18, 20, 21, 23,
24, 26, 27, 29, 32, 33, 35, 36, 38, 39,
41, 43, 45, 46, 49, 50, 52, 54, 55, 57,
58, 60, 61

base::rank, 43

class:matrix_OR_array_OR_table_OR_numeric
(MatrixGenerics-package), 3

colAlls, 4, 5

colAlls (rowAlls), 3

colAlls, ANY-method (rowAlls), 3

colAlls, matrix_OR_array_OR_table_OR_numeric-method
(rowAlls), 3

colAnyNAs, 6

colAnyNAs (rowAnyNAs), 5

colAnyNAs, ANY-method (rowAnyNAs), 5

colAnyNAs, matrix_OR_array_OR_table_OR_numeric-method
(rowAnyNAs), 5

colAnys, 7, 8

colAnys (rowAnys), 6

colAnys, ANY-method (rowAnys), 6

colAnys, matrix_OR_array_OR_table_OR_numeric-method
(rowAnys), 6

- colAvgPerRowSet, [9](#)
- colAvgPerRowSet (rowAvgPerColSet), [8](#)
- colAvgPerRowSet, ANY-method
(rowAvgPerColSet), [8](#)
- colAvgPerRowSet, matrix_OR_array_OR_table_OR_numeric-method
(rowAvgPerColSet), [8](#)
- colCollapse, [10, 11](#)
- colCollapse (rowCollapse), [10](#)
- colCollapse, ANY-method (rowCollapse), [10](#)
- colCollapse, matrix_OR_array_OR_table_OR_numeric-method
(rowCollapse), [10](#)
- colCounts, [12](#)
- colCounts (rowCounts), [11](#)
- colCounts, ANY-method (rowCounts), [11](#)
- colCounts, matrix_OR_array_OR_table_OR_numeric-method
(rowCounts), [11](#)
- colCummaxs, [14](#)
- colCummaxs (rowCummaxs), [13](#)
- colCummaxs, ANY-method (rowCummaxs), [13](#)
- colCummaxs, matrix_OR_array_OR_table_OR_numeric-method
(rowCummaxs), [13](#)
- colCummins, [15](#)
- colCummins (rowCummins), [14](#)
- colCummins, ANY-method (rowCummins), [14](#)
- colCummins, matrix_OR_array_OR_table_OR_numeric-method
(rowCummins), [14](#)
- colCumprods, [16, 17](#)
- colCumprods (rowCumprods), [16](#)
- colCumprods, ANY-method (rowCumprods), [16](#)
- colCumprods, matrix_OR_array_OR_table_OR_numeric-method
(rowCumprods), [16](#)
- colCumsums, [18](#)
- colCumsums (rowCumsums), [17](#)
- colCumsums, ANY-method (rowCumsums), [17](#)
- colCumsums, matrix_OR_array_OR_table_OR_numeric-method
(rowCumsums), [17](#)
- colDiffs, [20](#)
- colDiffs (rowDiffs), [19](#)
- colDiffs, ANY-method (rowDiffs), [19](#)
- colDiffs, matrix_OR_array_OR_table_OR_numeric-method
(rowDiffs), [19](#)
- colIQRDiffs, [21](#)
- colIQRDiffs (rowIQRDiffs), [20](#)
- colIQRDiffs, ANY-method (rowIQRDiffs), [20](#)
- colIQRDiffs, matrix_OR_array_OR_table_OR_numeric-method
(rowIQRDiffs), [20](#)
- colIQRs, [23](#)
- colIQRs (rowIQRs), [22](#)
- colIQRs, ANY-method (rowIQRs), [22](#)
- colIQRs, matrix_OR_array_OR_table_OR_numeric-method
(rowIQRs), [22](#)
- colLogSumExps, [24](#)
- colLogSumExps (rowLogSumExps), [23](#)
- colLogSumExps, ANY-method
(rowLogSumExps), [23](#)
- colLogSumExps, matrix_OR_array_OR_table_OR_numeric-method
(rowLogSumExps), [23](#)
- colMadDiffs, [26](#)
- colMadDiffs (rowMadDiffs), [25](#)
- colMadDiffs, ANY-method (rowMadDiffs), [25](#)
- colMadDiffs, matrix_OR_array_OR_table_OR_numeric-method
(rowMadDiffs), [25](#)
- colMads, [27, 28](#)
- colMads (rowMads), [26](#)
- colMads, ANY-method (rowMads), [26](#)
- colMads, matrix_OR_array_OR_table_OR_numeric-method
(rowMads), [26](#)
- colMaxs, [29](#)
- colMaxs (rowMaxs), [28](#)
- colMaxs, ANY-method (rowMaxs), [28](#)
- colMaxs, matrix_OR_array_OR_table_OR_numeric-method
(rowMaxs), [28](#)
- colMeans, [30](#)
- colMeans (rowMeans), [30](#)
- colMeans2, [32](#)
- colMeans2 (rowMeans2), [31](#)
- colMeans2, ANY-method (rowMeans2), [31](#)
- colMeans2, matrix_OR_array_OR_table_OR_numeric-method
(rowMeans2), [31](#)
- colMedians, [33](#)
- colMedians (rowMedians), [32](#)
- colMedians, ANY-method (rowMedians), [32](#)
- colMedians, matrix_OR_array_OR_table_OR_numeric-method
(rowMedians), [32](#)
- colMins, [35](#)
- colMins (rowMins), [34](#)
- colMins, ANY-method (rowMins), [34](#)
- colMins, matrix_OR_array_OR_table_OR_numeric-method
(rowMins), [34](#)
- colOrderStats, [36](#)
- colOrderStats (rowOrderStats), [35](#)
- colOrderStats, ANY-method
(rowOrderStats), [35](#)
- colOrderStats, matrix_OR_array_OR_table_OR_numeric-method
(rowOrderStats), [35](#)
- colProds, [38](#)

colProds (rowProds), [37](#)
 colProds, ANY-method (rowProds), [37](#)
 colProds, matrix_OR_array_OR_table_OR_numeric-method (rowProds), [37](#)
 colQuantiles, [39](#)
 colQuantiles (rowQuantiles), [38](#)
 colQuantiles, ANY-method (rowQuantiles), [38](#)
 colQuantiles, matrix_OR_array_OR_table_OR_numeric-method (rowQuantiles), [38](#)
 colRanges, [41](#)
 colRanges (rowRanges), [40](#)
 colRanges, ANY-method (rowRanges), [40](#)
 colRanges, matrix_OR_array_OR_table_OR_numeric-method (rowRanges), [40](#)
 colRanks, [43](#)
 colRanks (rowRanks), [42](#)
 colRanks, ANY-method (rowRanks), [42](#)
 colRanks, matrix_OR_array_OR_table_OR_numeric-method (rowRanks), [42](#)
 colSdDiffs, [45](#)
 colSdDiffs (rowSdDiffs), [44](#)
 colSdDiffs, ANY-method (rowSdDiffs), [44](#)
 colSdDiffs, matrix_OR_array_OR_table_OR_numeric-method (rowSdDiffs), [44](#)
 colSds, [46](#)
 colSds (rowSds), [45](#)
 colSds, ANY-method (rowSds), [45](#)
 colSds, matrix_OR_array_OR_table_OR_numeric-method (rowSds), [45](#)
 colSums, [47](#)
 colSums (rowSums), [47](#)
 colSums2, [49](#)
 colSums2 (rowSums2), [48](#)
 colSums2(), [38](#)
 colSums2, ANY-method (rowSums2), [48](#)
 colSums2, matrix_OR_array_OR_table_OR_numeric-method (rowSums2), [48](#)
 colTabulates, [50](#)
 colTabulates (rowTabulates), [49](#)
 colTabulates, ANY-method (rowTabulates), [49](#)
 colTabulates, matrix_OR_array_OR_table_OR_numeric-method (rowTabulates), [49](#)
 colVarDiffs, [52](#)
 colVarDiffs (rowVarDiffs), [51](#)
 colVarDiffs, ANY-method (rowVarDiffs), [51](#)
 colVarDiffs, matrix_OR_array_OR_table_OR_numeric-method (rowVarDiffs), [51](#)
 colVars, [54](#)
 colVars (rowVars), [53](#)
 colVars, ANY-method (rowVars), [53](#)
 colVars, matrix_OR_array_OR_table_OR_numeric-method (rowVars), [53](#)
 colWeightedMads, [55](#)
 colWeightedMads (rowWeightedMads), [54](#)
 colWeightedMads, ANY-method (rowWeightedMads), [54](#)
 colWeightedMads, matrix_OR_array_OR_table_OR_numeric-method (rowWeightedMads), [54](#)
 colWeightedMeans, [57](#)
 colWeightedMeans (rowWeightedMeans), [56](#)
 colWeightedMeans, ANY-method (rowWeightedMeans), [56](#)
 colWeightedMeans, matrix_OR_array_OR_table_OR_numeric-method (rowWeightedMeans), [56](#)
 colWeightedMedians, [58, 59](#)
 colWeightedMedians (rowWeightedMedians), [57](#)
 colWeightedMedians, ANY-method (rowWeightedMedians), [57](#)
 colWeightedMedians, matrix_OR_array_OR_table_OR_numeric-method (rowWeightedMedians), [57](#)
 colWeightedSds, [60](#)
 colWeightedSds (rowWeightedSds), [59](#)
 colWeightedSds, ANY-method (rowWeightedSds), [59](#)
 colWeightedSds, matrix_OR_array_OR_table_OR_numeric-method (rowWeightedSds), [59](#)
 colWeightedVars, [61](#)
 colWeightedVars (rowWeightedVars), [60](#)
 colWeightedVars, ANY-method (rowWeightedVars), [60](#)
 colWeightedVars, matrix_OR_array_OR_table_OR_numeric-method (rowWeightedVars), [60](#)
 cummax, [14](#)
 cummin, [15](#)
 cumprod, [17](#)
 cumsum, [18](#)
 diff, [20](#)
 FALSE, [4, 6, 7, 10, 12, 13, 15, 16, 18, 19, 21, 22, 24, 26, 27, 29, 31, 33, 34, 36, 37, 39, 41, 42, 44, 46, 49, 50, 52, 53, 55, 57, 58, 60, 61](#)

- integer, [4](#), [7](#), [9](#), [10](#), [12](#), [13](#), [15](#), [16](#), [18](#), [19](#), [24](#),
[27](#), [29](#), [31](#), [33](#), [34](#), [36](#), [41](#), [43](#), [46](#), [49](#),
[53](#)
- internal-helpers, [3](#)
- IQR, [23](#)
- is.na, [6](#)
- logical, [4](#), [6](#), [7](#)
- mad, [27](#), [55](#)
- matrix, [4](#), [6](#), [7](#), [9](#), [10](#), [12](#), [14–18](#), [20](#), [21](#), [23](#),
[24](#), [26](#), [27](#), [29](#), [32](#), [33](#), [35](#), [36](#), [38](#), [39](#),
[41](#), [43](#), [45](#), [46](#), [49](#), [50](#), [52](#), [54](#), [55](#), [57](#),
[58](#), [60](#), [61](#)
- matrix_OR_array_OR_table_OR_numeric
(MatrixGenerics-package), [3](#)
- matrix_OR_array_OR_table_OR_numeric-class
(MatrixGenerics-package), [3](#)
- MatrixGenerics-package, [3](#)
- NA, [4](#), [7](#), [12](#), [21](#), [22](#), [24](#), [26](#), [27](#), [29–31](#), [33](#), [34](#),
[37](#), [39](#), [41](#), [44](#), [46](#), [47](#), [49](#), [52](#), [53](#), [55](#),
[57](#), [58](#), [60](#), [61](#)
- names, [43](#)
- NaN, [4](#), [7](#), [12](#), [21](#), [22](#), [24](#), [26](#), [27](#), [29–31](#), [33](#), [34](#),
[37](#), [39](#), [41](#), [44](#), [46](#), [47](#), [49](#), [52](#), [53](#), [55](#),
[57](#), [58](#), [60](#), [61](#)
- normarg_center (internal-helpers), [3](#)
- NULL, [4](#), [6](#), [7](#), [9](#), [10](#), [12](#), [13](#), [15](#), [16](#), [18](#), [19](#), [21](#),
[22](#), [24](#), [25](#), [27](#), [29](#), [31](#), [33](#), [34](#), [36](#), [37](#),
[39](#), [40](#), [42](#), [44](#), [46](#), [48](#), [50](#), [52](#), [53](#), [55](#),
[57](#), [58](#), [60](#), [61](#)
- numeric, [4](#), [6](#), [7](#), [9–21](#), [23](#), [24](#), [26–36](#), [38](#), [39](#),
[41](#), [43](#), [45–47](#), [49](#), [50](#), [52–55](#), [57](#), [58](#),
[60](#), [61](#)
- prod, [38](#)
- quantile, [39](#)
- range, [41](#)
- RangedSummarizedExperiment, [41](#)
- rowAlls, [3](#), [4](#), [5](#), [8](#), [12](#)
- rowAlls, ANY-method (rowAlls), [3](#)
- rowAlls, matrix_OR_array_OR_table_OR_numeric-method
(rowAlls), [3](#)
- rowAnyNAs, [5](#), [6](#)
- rowAnyNAs, ANY-method (rowAnyNAs), [5](#)
- rowAnyNAs, matrix_OR_array_OR_table_OR_numeric-method
(rowAnyNAs), [5](#)
- rowAnys, [5](#), [6](#), [6](#), [7](#), [8](#), [12](#)
- rowAnys, ANY-method (rowAnys), [6](#)
- rowAnys, matrix_OR_array_OR_table_OR_numeric-method
(rowAnys), [6](#)
- rowAvgsPerColSet, [8](#), [9](#)
- rowAvgsPerColSet, ANY-method
(rowAvgsPerColSet), [8](#)
- rowAvgsPerColSet, matrix_OR_array_OR_table_OR_numeric-method
(rowAvgsPerColSet), [8](#)
- rowCollapse, [10](#), [10](#), [11](#)
- rowCollapse, ANY-method (rowCollapse), [10](#)
- rowCollapse, matrix_OR_array_OR_table_OR_numeric-method
(rowCollapse), [10](#)
- rowCounts, [11](#), [12](#)
- rowCounts, ANY-method (rowCounts), [11](#)
- rowCounts, matrix_OR_array_OR_table_OR_numeric-method
(rowCounts), [11](#)
- rowCummaxs, [13](#), [14](#)
- rowCummaxs, ANY-method (rowCummaxs), [13](#)
- rowCummaxs, matrix_OR_array_OR_table_OR_numeric-method
(rowCummaxs), [13](#)
- rowCummins, [14](#), [15](#)
- rowCummins, ANY-method (rowCummins), [14](#)
- rowCummins, matrix_OR_array_OR_table_OR_numeric-method
(rowCummins), [14](#)
- rowCumprods, [16](#), [16](#), [17](#)
- rowCumprods, ANY-method (rowCumprods), [16](#)
- rowCumprods, matrix_OR_array_OR_table_OR_numeric-method
(rowCumprods), [16](#)
- rowCumsums, [17](#), [18](#)
- rowCumsums, ANY-method (rowCumsums), [17](#)
- rowCumsums, matrix_OR_array_OR_table_OR_numeric-method
(rowCumsums), [17](#)
- rowDiffs, [19](#), [20](#)
- rowDiffs, ANY-method (rowDiffs), [19](#)
- rowDiffs, matrix_OR_array_OR_table_OR_numeric-method
(rowDiffs), [19](#)
- rowIQRDiffs, [20](#), [21](#)
- rowIQRDiffs, ANY-method (rowIQRDiffs), [20](#)
- rowIQRDiffs, matrix_OR_array_OR_table_OR_numeric-method
(rowIQRDiffs), [20](#)
- rowIQRs, [21](#), [22](#), [23](#)
- rowIQRs, ANY-method (rowIQRs), [22](#)
- rowIQRs, matrix_OR_array_OR_table_OR_numeric-method
(rowIQRs), [22](#)
- rowLogSumExps, [23](#), [24](#)
- rowLogSumExps, ANY-method
(rowLogSumExps), [23](#)

rowLogSumExps, matrix_OR_array_OR_table_OR_numeric-method
 (rowLogSumExps), 23
 rowMadDiffs, 25, 26
 rowMadDiffs, ANY-method (rowMadDiffs), 25
 rowMadDiffs, matrix_OR_array_OR_table_OR_numeric-method
 (rowMadDiffs), 25
 rowMads, 26, 27, 28, 55
 rowMads(), 23
 rowMads, ANY-method (rowMads), 26
 rowMads, matrix_OR_array_OR_table_OR_numeric-method
 (rowMads), 26
 rowMaxs, 14, 28, 29, 35, 41
 rowMaxs, ANY-method (rowMaxs), 28
 rowMaxs, matrix_OR_array_OR_table_OR_numeric-method
 (rowMaxs), 28
 rowMeans, 28, 30, 32, 33, 46, 49, 54
 rowMeans2, 28, 31, 32, 33, 46, 49, 54, 57
 rowMeans2, ANY-method (rowMeans2), 31
 rowMeans2, matrix_OR_array_OR_table_OR_numeric-method
 (rowMeans2), 31
 rowMedians, 32, 33, 59
 rowMedians, ANY-method (rowMedians), 32
 rowMedians, matrix_OR_array_OR_table_OR_numeric-method
 (rowMedians), 32
 rowMins, 15, 29, 34, 35, 41
 rowMins, ANY-method (rowMins), 34
 rowMins, matrix_OR_array_OR_table_OR_numeric-method
 (rowMins), 34
 rowOrderStats, 35, 36
 rowOrderStats, ANY-method
 (rowOrderStats), 35
 rowOrderStats, matrix_OR_array_OR_table_OR_numeric-method
 (rowOrderStats), 35
 rowProds, 37, 38
 rowProds, ANY-method (rowProds), 37
 rowProds, matrix_OR_array_OR_table_OR_numeric-method
 (rowProds), 37
 rowQuantiles, 38, 39
 rowQuantiles, ANY-method (rowQuantiles),
 38
 rowQuantiles, matrix_OR_array_OR_table_OR_numeric-method
 (rowQuantiles), 38
 rowRanges, 40, 41
 rowRanges, ANY-method (rowRanges), 40
 rowRanges, matrix_OR_array_OR_table_OR_numeric-method
 (rowRanges), 40
 rowRanks, 42, 43
 rowRanks, ANY-method (rowRanks), 42
 rowRanks, matrix_OR_array_OR_table_OR_numeric-method
 (rowRanks), 42
 rowSdDiffs, 44, 45
 rowSdDiffs, ANY-method (rowSdDiffs), 44
 rowSdDiffs, matrix_OR_array_OR_table_OR_numeric-method
 (rowSdDiffs), 44
 rowSds, 23, 28, 45, 46, 54, 60
 rowSds(), 45
 rowSds, ANY-method (rowSds), 45
 rowSds, matrix_OR_array_OR_table_OR_numeric-method
 (rowSds), 45
 rowSums, 47
 rowSums2, 38, 48, 49
 rowSums2(), 24
 rowSums2, ANY-method (rowSums2), 48
 rowSums2, matrix_OR_array_OR_table_OR_numeric-method
 (rowSums2), 48
 rowTabulates, 49, 50
 rowTabulates, ANY-method (rowTabulates),
 49
 rowTabulates, matrix_OR_array_OR_table_OR_numeric-method
 (rowTabulates), 49
 rowVarDiffs, 51, 52
 rowVarDiffs, ANY-method (rowVarDiffs), 51
 rowVarDiffs, matrix_OR_array_OR_table_OR_numeric-method
 (rowVarDiffs), 51
 rowVars, 32, 46, 53, 54, 62
 rowVars(), 52
 rowVars, ANY-method (rowVars), 53
 rowVars, matrix_OR_array_OR_table_OR_numeric-method
 (rowVars), 53
 rowWeightedMads, 54, 55
 rowWeightedMads, ANY-method
 (rowWeightedMads), 54
 rowWeightedMads, matrix_OR_array_OR_table_OR_numeric-method
 (rowWeightedMads), 54
 rowWeightedMeans, 56, 57
 rowWeightedMeans, ANY-method
 (rowWeightedMeans), 56
 rowWeightedMeans, matrix_OR_array_OR_table_OR_numeric-method
 (rowWeightedMeans), 56
 rowWeightedMedians, 57, 58, 59
 rowWeightedMedians, ANY-method
 (rowWeightedMedians), 57
 rowWeightedMedians, matrix_OR_array_OR_table_OR_numeric-method
 (rowWeightedMedians), 57
 rowWeightedSds, 59, 60
 rowWeightedSds, ANY-method

(rowWeightedSds), 59
rowWeightedSds,matrix_OR_array_OR_table_OR_numeric-method
 (rowWeightedSds), 59
rowWeightedVars, 60, 61
rowWeightedVars,ANY-method
 (rowWeightedVars), 60
rowWeightedVars,matrix_OR_array_OR_table_OR_numeric-method
 (rowWeightedVars), 60

stats::quantile, 39
sum, 49

table, 4, 6, 7, 9, 10, 12, 14–16, 18, 20, 21, 23,
 24, 26, 27, 29, 32, 33, 35, 36, 38, 39,
 41, 43, 45, 46, 49, 50, 52, 54, 55, 57,
 58, 60, 61
TRUE, 4, 6, 7, 10, 12, 13, 15, 16, 18, 19, 21, 22,
 24, 26, 27, 29–31, 33, 34, 36, 37, 39,
 41, 42, 44, 46, 47, 49, 50, 52, 53, 55,
 57, 58, 60, 61

var, 54
vector, 4, 6, 7, 9–13, 15, 16, 18, 19, 21–50,
 52–55, 57, 58, 60, 61