

# Package ‘MatrixGenerics’

October 17, 2020

**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.

**Version** 1.0.2

**Imports** methods

**Depends** matrixStats (>= 0.56.0)

**License** Artistic-2.0

**Encoding** UTF-8

**URL** <https://github.com/Bioconductor/MatrixGenerics>

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

**RoxygenNote** 7.1.0

**Roxygen** list(markdown = TRUE, old\_usage = TRUE)

**Suggests** testthat (>= 2.1.0)

**biocViews** Infrastructure, Software

**git\_url** <https://git.bioconductor.org/packages/MatrixGenerics>

**git\_branch** RELEASE\_3\_11

**git\_last\_commit** a221e4c

**git\_last\_commit\_date** 2020-05-23

**Date/Publication** 2020-10-16

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

**Maintainer** Peter Hickey <[peter.hickey@gmail.com](mailto:peter.hickey@gmail.com)>

**R topics documented:**

rowAlls . . . . .	2
rowAnyNAs . . . . .	4
rowAnys . . . . .	5
rowAvgsPerColSet . . . . .	7
rowCollapse . . . . .	8
rowCounts . . . . .	10
rowCummaxs . . . . .	11
rowCummins . . . . .	13
rowCumprods . . . . .	14
rowCumsums . . . . .	16
rowDiffs . . . . .	17
rowIQRDiffs . . . . .	19
rowIQRs . . . . .	20
rowLogSumExps . . . . .	21
rowMadDiffs . . . . .	23
rowMads . . . . .	24
rowMaxs . . . . .	26
rowMeans2 . . . . .	27
rowMedians . . . . .	29
rowMins . . . . .	30
rowOrderStats . . . . .	32
rowProds . . . . .	33
rowQuantiles . . . . .	34
rowRanges . . . . .	36
rowRanks . . . . .	38
rowSdDiffs . . . . .	39
rowSds . . . . .	41
rowSums2 . . . . .	43
rowTabulates . . . . .	44
rowVarDiffs . . . . .	45
rowVars . . . . .	47
rowWeightedMads . . . . .	49
rowWeightedMeans . . . . .	50
rowWeightedMedians . . . . .	52
rowWeightedSds . . . . .	53
rowWeightedVars . . . . .	54
<b>Index</b>	<b>57</b>

rowAlls

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

**Description**

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

**Usage**

```

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

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

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

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

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

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

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

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

```

**Arguments**

<code>x</code>	An $N \times K$ matrix-like object.
<code>rows, cols</code>	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
<code>value</code>	The value to search for.
<code>na.rm</code>	If <a href="#">TRUE</a> , <a href="#">NAs</a> are excluded first, otherwise not.
<code>...</code>	Additional arguments passed to specific methods.
<code>dim.</code>	An <a href="#">integer vector</a> of length two specifying the dimension of <code>x</code> , essential when <code>x</code> is a <a href="#">numeric</a> 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](#), or [numeric](#) call `matrixStats::rowAlls/matrixStats::colAlls`.

**Value**

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

**See Also**

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

**Examples**

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

print(mat)

rowAlls(mat)
colAlls(mat)
```

---

rowAnyNAs

---

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


---

**Description**

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

**Usage**

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

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

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

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

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

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

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

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

**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.

**Details**

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

**Value**

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

**See Also**

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

**Examples**

```
mat <- matrix(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, ...)

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

## S4 method for signature 'numeric'
```

```

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

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

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

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

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

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

```

### Arguments

<code>x</code>	An $N \times K$ matrix-like object.
<code>rows, cols</code>	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
<code>value</code>	The value to search for.
<code>na.rm</code>	If <a href="#">TRUE</a> , <a href="#">NAs</a> are excluded first, otherwise not.
<code>...</code>	Additional arguments passed to specific methods.
<code>dim.</code>	An <a href="#">integer vector</a> of length two specifying the dimension of <code>x</code> , essential when <code>x</code> is a <a href="#">numeric</a> 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](#), or [numeric](#) call `matrixStats::rowAnys` / `matrixStats::colAnys`.

### Value

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

### See Also

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

### Examples

```

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

```

```

mat[4, 1] <- 0

print(mat)

rowAnys(mat)
colAnys(mat)

```

---

rowAvsPerColSet	<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, ...,
  tFUN = FALSE)

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

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

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

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

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

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

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

```

### Arguments

X                    An NxM matrix-like object.

W	An optional numeric NxM matrix of weights.
rows, cols	A <a href="#">vector</a> indicating the subset (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
S	An <a href="#">integer</a> 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 each column (row) subset of X that is specified by S.
...	Additional arguments passed to FUN.
tFUN	If TRUE, X is transposed before it is passed to FUN.

### Details

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

### Value

Returns a numeric JxN (MxJ) matrix.

### See Also

- `matrixStats::rowAvsPerColSet()` and `matrixStats::colAvsPerRowSet()` 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)

rowAvsPerColSet(mat, S = S, FUN = rowMeans)
rowAvsPerColSet(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, ...)

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

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

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

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

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

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

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

```

**Arguments**

x	An NxK matrix-like object.
idxs	An index <a href="#">vector</a> with the position to extract. It is recycled to match the number of rows (column)
rows, cols	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
...	Additional arguments passed to specific methods.
dim.	An <a href="#">integer vector</a> of length two specifying the dimension of x, essential when x is a <a href="#">numeric</a> 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](#), 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

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

---

**Description**

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

**Usage**

```

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

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

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

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

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

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

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

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

```

**Arguments**

x	An NxK matrix-like object.
rows, cols	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
value	The value to search for.
na.rm	If <a href="#">TRUE</a> , <a href="#">NAs</a> are excluded first, otherwise not.
...	Additional arguments passed to specific methods.
dim.	An <a href="#">integer vector</a> of length two specifying the dimension of x, essential when x is a <a href="#">numeric</a> 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](#), 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

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

---

**Description**

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

**Usage**

```

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

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

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

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

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

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

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

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

```

**Arguments**

<code>x</code>	An NxK matrix-like object.
<code>rows, cols</code>	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <code>NULL</code> , no subsetting is done.
<code>...</code>	Additional arguments passed to specific methods.
<code>dim.</code>	An <a href="#">integer vector</a> of length two specifying the dimension of <code>x</code> , essential when <code>x</code> is a <a href="#">numeric</a> 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](#), 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	<i>Calculates the cumulative minima for each row (column) of a matrix-like object</i>
------------	---

---

**Description**

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

**Usage**

```

rowCummins(x, rows = NULL, cols = NULL, ...)

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

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

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

colCummins(x, rows = NULL, cols = NULL, ...)

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

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

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

```

**Arguments**

x	An NxK matrix-like object.
rows, cols	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
...	Additional arguments passed to specific methods.

`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**, 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, ...)

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

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

## S4 method for signature 'array'
```

```

rowCumprods(x, rows = NULL, cols = NULL, dim. = dim(x), ...)

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

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

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

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

```

### Arguments

<code>x</code>	An <code>NxK</code> matrix-like object.
<code>rows, cols</code>	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <code>NULL</code> , no subsetting is done.
<code>...</code>	Additional arguments passed to specific methods.
<code>dim.</code>	An <a href="#">integer vector</a> of length two specifying the dimension of <code>x</code> , essential when <code>x</code> is a <a href="#">numeric</a> 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](#), 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, ...)

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

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

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

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

## S4 method for signature 'matrix'
colCumsums(x, rows = NULL, cols = NULL, dim. = dim(x), ...)

## S4 method for signature 'numeric'
colCumsums(x, rows = NULL, cols = NULL, dim. = dim(x), ...)

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

### Arguments

x	An NxK matrix-like object.
rows, cols	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
...	Additional arguments passed to specific methods.
dim.	An <a href="#">integer vector</a> of length two specifying the dimension of x, essential when x is a <a href="#">numeric</a> 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](#), 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, ...)

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

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

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

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

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

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

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

### Arguments

x	An NxK matrix-like object.
rows, cols	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
lag	An integer specifying the lag.
differences	An integer specifying the order of difference.
...	Additional arguments passed to specific methods.
dim.	An <a href="#">integer vector</a> of length two specifying the dimension of x, essential when x is a <a href="#">numeric</a> 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](#), 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	<i>Calculates the interquartile range of the difference between each element of a row (column) of a matrix-like object</i>
-------------	--

---

### 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, ...)

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

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

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

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

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

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

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

### Arguments

<code>x</code>	An $N \times K$ matrix-like object.
<code>rows, cols</code>	A <a href="#">vector</a> 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> , <code>NA</code> s are excluded first, otherwise not.
<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.

**Details**

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

**Value**

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

**See Also**

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

**Examples**

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

print(mat)

rowIQRDiffs(mat)
colIQRDiffs(mat)
```

---

rowIQRs

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

---

**Description**

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

**Usage**

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

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

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

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

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

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

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

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

### Arguments

`x` An  $N \times K$  matrix-like object.

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

`na.rm` If `TRUE`, `NA`s are excluded first, otherwise not.

`...` Additional arguments passed to specific methods.

### Details

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

### Value

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

### See Also

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

### Examples

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

print(mat)

rowIQRs(mat)
colIQRs(mat)
```

---

rowLogSumExps

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

---

### Description

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

**Usage**

```

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

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

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

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

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

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

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

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

```

**Arguments**

<code>lx</code>	An NxK matrix-like object. Typically <code>lx</code> are $\log(x)$ values.
<code>rows, cols</code>	A <a href="#">vector</a> 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> , <code>NA</code> s are excluded first, otherwise not.
<code>...</code>	Additional arguments passed to specific methods.
<code>dim.</code>	An <a href="#">integer vector</a> of length two specifying the dimension of <code>x</code> , essential when <code>x</code> is a <a href="#">numeric</a> 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](#), 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

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


---

**Description**

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

**Usage**

```

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

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

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

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

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

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

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

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

```

**Arguments**

<code>x</code>	An NxK matrix-like object.
<code>rows, cols</code>	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
<code>na.rm</code>	If <a href="#">TRUE</a> , NAs are excluded first, otherwise not.
<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) x before estimation.
<code>...</code>	Additional arguments passed to specific methods.

**Details**

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

**Value**

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

**See Also**

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

**Examples**

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

print(mat)

rowMadDiffs(mat)
colMadDiffs(mat)
```

---

rowMads

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

---

**Description**

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



**Usage**

```

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

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

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

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

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

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

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

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

```

**Arguments**

<code>x</code>	An $N \times K$ matrix-like object.
<code>rows, cols</code>	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <code>NULL</code> , no subsetting is done.
<code>center</code>	(optional) the center, defaults to the row means
<code>constant</code>	A scale factor. See <code>stats::mad()</code> for details.
<code>na.rm</code>	If <code>TRUE</code> , <code>NA</code> s are excluded first, otherwise not.
<code>...</code>	Additional arguments passed to specific methods.
<code>dim.</code>	An <a href="#">integer vector</a> of length two specifying the dimension of <code>x</code> , essential when <code>x</code> is a <a href="#">numeric</a> 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`, 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, ...)

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

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

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

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

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

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

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

**Arguments**

x	An NxK matrix-like object.
rows, cols	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
na.rm	If <a href="#">TRUE</a> , <a href="#">NAs</a> are excluded first, otherwise not.
...	Additional arguments passed to specific methods.
dim.	An <a href="#">integer vector</a> of length two specifying the dimension of x, essential when x is a <a href="#">numeric</a> 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](#), 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)
```

---

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, ...)

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

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

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

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

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

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

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

```

**Arguments**

<code>x</code>	An $N \times K$ matrix-like object.
<code>rows, cols</code>	A <a href="#">vector</a> 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> , <code>NA</code> s are excluded first, otherwise not.
<code>...</code>	Additional arguments passed to specific methods.
<code>dim.</code>	An <a href="#">integer vector</a> of length two specifying the dimension of <code>x</code> , essential when <code>x</code> is a <a href="#">numeric</a> 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](#), 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	<i>Calculates the median for each row (column) of a matrix-like object</i>
------------	--

---

**Description**

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

**Usage**

```

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

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

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

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

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

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

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

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

```

**Arguments**

<code>x</code>	An NxK matrix-like object.
<code>rows, cols</code>	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
<code>na.rm</code>	If <a href="#">TRUE</a> , NAs are excluded first, otherwise not.
<code>...</code>	Additional arguments passed to specific methods.

`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**, 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, ...)

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

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

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

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

### Arguments

x	An NxK matrix-like object.
rows, cols	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
na.rm	If <a href="#">TRUE</a> , <a href="#">NAs</a> are excluded first, otherwise not.
...	Additional arguments passed to specific methods.
dim.	An <a href="#">integer vector</a> of length two specifying the dimension of x, essential when x is a <a href="#">numeric</a> 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](#), 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, ...)
```

## S4 method for signature 'matrix'

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

## S4 method for signature 'numeric'

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

## S4 method for signature 'array'

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

colOrderStats(x, rows = NULL, cols = NULL, which, ...)

## S4 method for signature 'matrix'

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

## S4 method for signature 'numeric'

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

## S4 method for signature 'array'

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

### Arguments

x	An NxK matrix-like object.
rows, cols	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
which	An integer index in [1,K] ([1,N]) indicating which order statistic to be returned
...	Additional arguments passed to specific methods.
dim.	An <a href="#">integer vector</a> of length two specifying the dimension of x, essential when x is a <a href="#">numeric</a> 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](#), 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

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


---

**Description**

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

**Usage**

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

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

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

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

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

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

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

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

**Arguments**

x	An NxK matrix-like object.
rows, cols	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
na.rm	If <a href="#">TRUE</a> , <a href="#">NAs</a> are excluded first, otherwise not.
...	Additional arguments passed to specific methods.
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](#), 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, ...)

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

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

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

colQuantiles(x, rows = NULL, cols = NULL, probs = seq(from = 0, to = 1,
  by = 0.25), na.rm = FALSE, ...)

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

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

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

```

**Arguments**

x	An NxK matrix-like object.
rows, cols	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
probs	A numeric vector of J probabilities in [0, 1].
na.rm	If <a href="#">TRUE</a> , <a href="#">NAs</a> are excluded first, otherwise not.
...	Additional arguments passed to specific methods.
type	An integer specifying the type of estimator. See <code>stats::quantile()</code> . for more details. Note, that this is not a generic argument and not all implementation of this function have to provide it.
drop	If <a href="#">TRUE</a> a vector is returned if $J == 1$ . Note, that this is not a generic argument and not all implementation of this function have to provide it.

**Details**

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

**Value**

a `numeric`  $N \times J$  ( $K \times J$ ) `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

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

---

**Description**

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

**Usage**

```
rowRanges(x, rows = NULL, cols = NULL, na.rm = FALSE, ...)

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

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

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

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

## S4 method for signature 'matrix'
```

```
colRanges(x, rows = NULL, cols = NULL, na.rm = FALSE, dim. = dim(x), ...)

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

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

### Arguments

<code>x</code>	An $N \times K$ matrix-like object.
<code>rows, cols</code>	A <a href="#">vector</a> 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> , <code>NA</code> s are excluded first, otherwise not.
<code>...</code>	Additional arguments passed to specific methods.
<code>dim.</code>	An <a href="#">integer vector</a> of length two specifying the dimension of <code>x</code> , essential when <code>x</code> is a <a href="#">numeric</a> 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`, or `numeric` call `matrixStats::rowRanges/matrixStats::colRanges`.

### Value

a [numeric](#)  $N \times 2$  ( $K \times 2$ ) [matrix](#), where  $N$  ( $K$ ) is the number of rows (columns) for which the ranges are calculated.

### 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"), ...)

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

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

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

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

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

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

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

### Arguments

x	An NxK matrix-like object.
rows, cols	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , 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.
dim.	An <a href="#">integer vector</a> of length two specifying the dimension of x, essential when x is a <a href="#">numeric</a> 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](#), 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"`. It has dimensions `N x J` (`K x J`) [matrix](#), where N (K) is the number of rows (columns) of the input x.

### See Also

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

### Examples

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

print(mat)

rowRanks(mat)
colRanks(mat)
```

---

rowSdDiffs

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

---

### Description

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

**Usage**

```

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

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

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

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

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

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

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

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

```

**Arguments**

<code>x</code>	An NxK matrix-like object.
<code>rows, cols</code>	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
<code>na.rm</code>	If <a href="#">TRUE</a> , NAs are excluded first, otherwise not.
<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) x before estimation.
<code>...</code>	Additional arguments passed to specific methods.

**Details**

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

**Value**

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



**See Also**

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

**Examples**

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

print(mat)

rowSdDiffs(mat)
colSdDiffs(mat)
```

---

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, ...)

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

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

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

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

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

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

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

### Arguments

<code>x</code>	An NxK matrix-like object.
<code>rows, cols</code>	A <a href="#">vector</a> 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> , NAs are excluded first, otherwise not.
<code>center</code>	(optional) the center, defaults to the row means
<code>...</code>	Additional arguments passed to specific methods.
<code>dim.</code>	An <a href="#">integer vector</a> of length two specifying the dimension of <code>x</code> , essential when <code>x</code> is a <a href="#">numeric</a> 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](#), 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)
```

---

rowSums2	<i>Calculates the sum for each row (column) of a matrix-like object</i>
----------	---

---

### Description

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

### Usage

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

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

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

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

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

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

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

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

### Arguments

x	An NxK matrix-like object.
rows, cols	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
na.rm	If <a href="#">TRUE</a> , <a href="#">NAs</a> are excluded first, otherwise not.
...	Additional arguments passed to specific methods.
dim.	An <a href="#">integer vector</a> of length two specifying the dimension of x, essential when x is a <a href="#">numeric</a> 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](#), 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, ...)

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

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

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

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

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

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

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

**Arguments**

x	An NxK matrix-like object.
rows, cols	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
values	the values to search for.
...	Additional arguments passed to specific methods.

**Details**

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

**Value**

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

**See Also**

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

**Examples**

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

print(mat)

rowTabulates(mat)
colTabulates(mat)

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

---

rowVarDiffs

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

---

**Description**

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

**Usage**

```

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

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

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

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

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

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

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

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

```

**Arguments**

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

**Details**

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

**Value**

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

**See Also**

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

**Examples**

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

print(mat)

rowVarDiffs(mat)
colVarDiffs(mat)
```

rowVars

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

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

**Usage**

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

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

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

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

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

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

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

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

### Arguments

<code>x</code>	An $N \times K$ matrix-like object.
<code>rows, cols</code>	A <a href="#">vector</a> 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> , <code>NA</code> s are excluded first, otherwise not.
<code>center</code>	(optional) the center, defaults to the row means.
<code>...</code>	Additional arguments passed to specific methods.
<code>dim.</code>	An <a href="#">integer vector</a> of length two specifying the dimension of <code>x</code> , essential when <code>x</code> is a <a href="#">numeric vector</a> . 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](#), 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, ...)
```

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

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

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

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

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

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

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

### Arguments

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

**Details**

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

**Value**

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

**See Also**

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

**Examples**

```
mat <- matrix(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)
```

---

<code>rowWeightedMeans</code>	<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, ...)

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

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

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

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

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

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

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

### Arguments

<code>x</code>	An $N \times K$ matrix-like object.
<code>w</code>	A <a href="#">numeric</a> vector of length $K$ ( $N$ ) that specifies by how much each element is weighted.
<code>rows, cols</code>	A <a href="#">vector</a> indicating the subset of rows (and/or columns) to operate over. If <a href="#">NULL</a> , no subsetting is done.
<code>na.rm</code>	If <a href="#">TRUE</a> , <a href="#">NAs</a> are excluded first, otherwise not.
<code>...</code>	Additional arguments passed to specific methods.

### Details

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

### Value

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

### See Also

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

### Examples

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

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

---

rowWeightedMedians	<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, ...)

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

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

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

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

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

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

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

### Arguments

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

### Details

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

### Value

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

**See Also**

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

**Examples**

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

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

---

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, ...)

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

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

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

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

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

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

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

**Arguments**

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

**Details**

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

**Value**

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

**See Also**

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

**Examples**

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

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

---

rowWeightedVars	<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, ...)

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

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

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

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

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

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

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

```

**Arguments**

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

**Details**

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

**Value**

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

**See Also**

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

**Examples**

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

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



# Index

## \* array

- rowAvgPerColSet, 7
- rowCollapse, 8
- rowCummaxs, 11
- rowCummins, 13
- rowCumprods, 14
- rowCumsums, 16
- rowDiffs, 17
- rowIQRDiffs, 19
- rowIQRs, 20
- rowLogSumExps, 21
- rowMadDiffs, 23
- rowMads, 24
- rowMaxs, 26
- rowMeans2, 27
- rowMedians, 29
- rowMins, 30
- rowOrderStats, 32
- rowProds, 33
- rowQuantiles, 34
- rowRanges, 36
- rowRanks, 38
- rowSdDiffs, 39
- rowSds, 41
- rowSums2, 43
- rowTabulates, 44
- rowVarDiffs, 45
- rowVars, 47
- rowWeightedMads, 49
- rowWeightedMeans, 50
- rowWeightedMedians, 52
- rowWeightedSds, 53
- rowWeightedVars, 54

## \* iteration

- rowAvgPerColSet, 7
- rowCollapse, 8
- rowCummaxs, 11
- rowCummins, 13
- rowCumprods, 14
- rowCumsums, 16
- rowDiffs, 17
- rowIQRDiffs, 19
- rowIQRs, 20

- rowLogSumExps, 21
- rowMadDiffs, 23
- rowMads, 24
- rowMaxs, 26
- rowMeans2, 27
- rowMedians, 29
- rowMins, 30
- rowOrderStats, 32
- rowProds, 33
- rowQuantiles, 34
- rowRanges, 36
- rowRanks, 38
- rowSdDiffs, 39
- rowSds, 41
- rowSums2, 43
- rowTabulates, 44
- rowVarDiffs, 45
- rowVars, 47
- rowWeightedMads, 49
- rowWeightedMeans, 50
- rowWeightedMedians, 52
- rowWeightedSds, 53
- rowWeightedVars, 54

## \* robust

- rowAvgPerColSet, 7
- rowCollapse, 8
- rowCummaxs, 11
- rowCummins, 13
- rowCumprods, 14
- rowCumsums, 16
- rowDiffs, 17
- rowIQRDiffs, 19
- rowIQRs, 20
- rowLogSumExps, 21
- rowMadDiffs, 23
- rowMads, 24
- rowMaxs, 26
- rowMeans2, 27
- rowMedians, 29
- rowMins, 30
- rowOrderStats, 32
- rowProds, 33
- rowQuantiles, 34

- rowRanges, 36
- rowRanks, 38
- rowSdDiffs, 39
- rowSds, 41
- rowSums2, 43
- rowTabulates, 44
- rowVarDiffs, 45
- rowVars, 47
- rowWeightedMads, 49
- rowWeightedMeans, 50
- rowWeightedMedians, 52
- rowWeightedSds, 53
- rowWeightedVars, 54
- \* **univar2**
  - rowAvsPerColSet, 7
- \* **univar**
  - rowCollapse, 8
  - rowCummaxs, 11
  - rowCummins, 13
  - rowCumprods, 14
  - rowCumsums, 16
  - rowDiffs, 17
  - rowIQRDiffs, 19
  - rowIQRs, 20
  - rowLogSumExps, 21
  - rowMadDiffs, 23
  - rowMads, 24
  - rowMaxs, 26
  - rowMeans2, 27
  - rowMedians, 29
  - rowMins, 30
  - rowOrderStats, 32
  - rowProds, 33
  - rowSdDiffs, 39
  - rowSds, 41
  - rowSums2, 43
  - rowTabulates, 44
  - rowVarDiffs, 45
  - rowVars, 47
  - rowWeightedMads, 49
  - rowWeightedMeans, 50
  - rowWeightedMedians, 52
  - rowWeightedSds, 53
  - rowWeightedVars, 54
- all, 4
- any, 5, 6
- array, 3–6, 8, 9, 11, 12, 14–16, 18, 20–22, 24, 25, 27, 28, 30–32, 34, 36, 37, 39, 40, 42, 43, 45, 46, 48, 50–52, 54, 55
- base::rank, 39
- colAlls, 4
- colAlls (rowAlls), 2
- colAlls, array-method (rowAlls), 2
- colAlls, matrix-method (rowAlls), 2
- colAlls, numeric-method (rowAlls), 2
- colAnyNAs, 5
- colAnyNAs (rowAnyNAs), 4
- colAnyNAs, array-method (rowAnyNAs), 4
- colAnyNAs, matrix-method (rowAnyNAs), 4
- colAnyNAs, numeric-method (rowAnyNAs), 4
- colAnys, 6
- colAnys (rowAnys), 5
- colAnys, array-method (rowAnys), 5
- colAnys, matrix-method (rowAnys), 5
- colAnys, numeric-method (rowAnys), 5
- colAvsPerRowSet, 8
- colAvsPerRowSet (rowAvsPerColSet), 7
- colAvsPerRowSet, array-method (rowAvsPerColSet), 7
- colAvsPerRowSet, matrix-method (rowAvsPerColSet), 7
- colAvsPerRowSet, numeric-method (rowAvsPerColSet), 7
- colCollapse, 9
- colCollapse (rowCollapse), 8
- colCollapse, array-method (rowCollapse), 8
- colCollapse, matrix-method (rowCollapse), 8
- colCollapse, numeric-method (rowCollapse), 8
- colCounts, 11
- colCounts (rowCounts), 10
- colCounts, array-method (rowCounts), 10
- colCounts, matrix-method (rowCounts), 10
- colCounts, numeric-method (rowCounts), 10
- colCummaxs, 12
- colCummaxs (rowCummaxs), 11
- colCummaxs, array-method (rowCummaxs), 11
- colCummaxs, matrix-method (rowCummaxs), 11
- colCummaxs, numeric-method (rowCummaxs), 11
- colCummins, 14
- colCummins (rowCummins), 13
- colCummins, array-method (rowCummins), 13
- colCummins, matrix-method (rowCummins), 13
- colCummins, numeric-method (rowCummins), 13
- colCumprods, 15
- colCumprods (rowCumprods), 14

- colCumprods, array-method (rowCumprods), 14
- colCumprods, matrix-method (rowCumprods), 14
- colCumprods, numeric-method (rowCumprods), 14
- colCumsums, 17
- colCumsums (rowCumsums), 16
- colCumsums, array-method (rowCumsums), 16
- colCumsums, matrix-method (rowCumsums), 16
- colCumsums, numeric-method (rowCumsums), 16
- colDiffs, 18
- colDiffs (rowDiffs), 17
- colDiffs, array-method (rowDiffs), 17
- colDiffs, matrix-method (rowDiffs), 17
- colDiffs, numeric-method (rowDiffs), 17
- colIQRDiffs, 20
- colIQRDiffs (rowIQRDiffs), 19
- colIQRDiffs, array-method (rowIQRDiffs), 19
- colIQRDiffs, matrix-method (rowIQRDiffs), 19
- colIQRDiffs, numeric-method (rowIQRDiffs), 19
- colIQRs, 21
- colIQRs (rowIQRs), 20
- colIQRs, array-method (rowIQRs), 20
- colIQRs, matrix-method (rowIQRs), 20
- colIQRs, numeric-method (rowIQRs), 20
- colLogSumExps, 22
- colLogSumExps (rowLogSumExps), 21
- colLogSumExps, array-method (rowLogSumExps), 21
- colLogSumExps, matrix-method (rowLogSumExps), 21
- colLogSumExps, numeric-method (rowLogSumExps), 21
- colMadDiffs, 24
- colMadDiffs (rowMadDiffs), 23
- colMadDiffs, array-method (rowMadDiffs), 23
- colMadDiffs, matrix-method (rowMadDiffs), 23
- colMadDiffs, numeric-method (rowMadDiffs), 23
- colMads, 26
- colMads (rowMads), 24
- colMads, array-method (rowMads), 24
- colMads, matrix-method (rowMads), 24
- colMads, numeric-method (rowMads), 24
- colMaxs, 27
- colMaxs (rowMaxs), 26
- colMaxs, array-method (rowMaxs), 26
- colMaxs, matrix-method (rowMaxs), 26
- colMaxs, numeric-method (rowMaxs), 26
- colMeans2, 28
- colMeans2 (rowMeans2), 27
- colMeans2, array-method (rowMeans2), 27
- colMeans2, matrix-method (rowMeans2), 27
- colMeans2, numeric-method (rowMeans2), 27
- colMedians, 30
- colMedians (rowMedians), 29
- colMedians, array-method (rowMedians), 29
- colMedians, matrix-method (rowMedians), 29
- colMedians, numeric-method (rowMedians), 29
- colMins, 31
- colMins (rowMins), 30
- colMins, array-method (rowMins), 30
- colMins, matrix-method (rowMins), 30
- colMins, numeric-method (rowMins), 30
- colOrderStats, 33
- colOrderStats (rowOrderStats), 32
- colOrderStats, array-method (rowOrderStats), 32
- colOrderStats, matrix-method (rowOrderStats), 32
- colOrderStats, numeric-method (rowOrderStats), 32
- colProds, 34
- colProds (rowProds), 33
- colProds, array-method (rowProds), 33
- colProds, matrix-method (rowProds), 33
- colProds, numeric-method (rowProds), 33
- colQuantiles, 36
- colQuantiles (rowQuantiles), 34
- colQuantiles, array-method (rowQuantiles), 34
- colQuantiles, matrix-method (rowQuantiles), 34
- colQuantiles, numeric-method (rowQuantiles), 34
- colRanges, 37
- colRanges (rowRanges), 36
- colRanges, array-method (rowRanges), 36
- colRanges, matrix-method (rowRanges), 36
- colRanges, numeric-method (rowRanges), 36
- colRanks, 39
- colRanks (rowRanks), 38
- colRanks, array-method (rowRanks), 38
- colRanks, matrix-method (rowRanks), 38

- colRanks, numeric-method (rowRanks), 38
- colSdDiffs, 41
- colSdDiffs (rowSdDiffs), 39
- colSdDiffs, array-method (rowSdDiffs), 39
- colSdDiffs, matrix-method (rowSdDiffs), 39
- colSdDiffs, numeric-method (rowSdDiffs), 39
- colSds, 42
- colSds (rowSds), 41
- colSds, array-method (rowSds), 41
- colSds, matrix-method (rowSds), 41
- colSds, numeric-method (rowSds), 41
- colSums2, 44
- colSums2 (rowSums2), 43
- colSums2(), 34
- colSums2, array-method (rowSums2), 43
- colSums2, matrix-method (rowSums2), 43
- colSums2, numeric-method (rowSums2), 43
- colTabulates, 45
- colTabulates (rowTabulates), 44
- colTabulates, array-method (rowTabulates), 44
- colTabulates, matrix-method (rowTabulates), 44
- colTabulates, numeric-method (rowTabulates), 44
- colVarDiffs, 47
- colVarDiffs (rowVarDiffs), 45
- colVarDiffs, array-method (rowVarDiffs), 45
- colVarDiffs, matrix-method (rowVarDiffs), 45
- colVarDiffs, numeric-method (rowVarDiffs), 45
- colVars, 48
- colVars (rowVars), 47
- colVars, array-method (rowVars), 47
- colVars, matrix-method (rowVars), 47
- colVars, numeric-method (rowVars), 47
- colWeightedMads, 50
- colWeightedMads (rowWeightedMads), 49
- colWeightedMads, array-method (rowWeightedMads), 49
- colWeightedMads, matrix-method (rowWeightedMads), 49
- colWeightedMads, numeric-method (rowWeightedMads), 49
- colWeightedMeans, 51
- colWeightedMeans (rowWeightedMeans), 50
- colWeightedMeans, array-method (rowWeightedMeans), 50
- colWeightedMeans, matrix-method (rowWeightedMeans), 50
- colWeightedMeans, numeric-method (rowWeightedMeans), 50
- colWeightedMedians, 53
- colWeightedMedians (rowWeightedMedians), 52
- colWeightedMedians, array-method (rowWeightedMedians), 52
- colWeightedMedians, matrix-method (rowWeightedMedians), 52
- colWeightedMedians, numeric-method (rowWeightedMedians), 52
- colWeightedSds, 54
- colWeightedSds (rowWeightedSds), 53
- colWeightedSds, array-method (rowWeightedSds), 53
- colWeightedSds, matrix-method (rowWeightedSds), 53
- colWeightedSds, numeric-method (rowWeightedSds), 53
- colWeightedVars, 55
- colWeightedVars (rowWeightedVars), 54
- colWeightedVars, array-method (rowWeightedVars), 54
- colWeightedVars, matrix-method (rowWeightedVars), 54
- colWeightedVars, numeric-method (rowWeightedVars), 54
- cummax, 12
- cummin, 14
- cumprod, 15
- cumsum, 17
- diff, 18
- integer, 3, 6, 8, 9, 11, 12, 14–16, 18, 22, 25, 27, 28, 30–32, 37, 39, 42, 43, 48
- IQR, 21
- is.na, 5
- logical, 3, 5, 6
- mad, 25, 49
- matrix, 3–6, 8, 9, 11, 12, 14–16, 18, 20–22, 24, 25, 27, 28, 30–32, 34, 36, 37, 39, 40, 42, 43, 45, 46, 48, 50–52, 54, 55
- NA, 3, 6, 11, 19, 21, 22, 24, 25, 27–29, 31, 34, 35, 37, 40, 42, 43, 46, 48, 49, 51, 52, 54, 55
- NULL, 3, 5, 6, 8, 9, 11–13, 15, 16, 18, 19, 21, 22, 24, 25, 27–29, 31, 32, 34, 35, 37,

- 38, 40, 42, 43, 45, 46, 48, 49, 51, 52, 54, 55
- numeric, 3–6, 8, 9, 11, 12, 14–16, 18, 20–22, 24, 25, 27, 28, 30–32, 34, 36, 37, 39, 40, 42, 43, 45, 46, 48–52, 54, 55
- prod, 34
- quantile, 35
- range, 37
- rowAlls, 2, 4, 11
- rowAlls, array-method (rowAlls), 2
- rowAlls, matrix-method (rowAlls), 2
- rowAlls, numeric-method (rowAlls), 2
- rowAnyNAs, 4, 5
- rowAnyNAs, array-method (rowAnyNAs), 4
- rowAnyNAs, matrix-method (rowAnyNAs), 4
- rowAnyNAs, numeric-method (rowAnyNAs), 4
- rowAnys, 4, 5, 5, 6, 11
- rowAnys, array-method (rowAnys), 5
- rowAnys, matrix-method (rowAnys), 5
- rowAnys, numeric-method (rowAnys), 5
- rowAvgPerColSet, 7, 8
- rowAvgPerColSet, array-method (rowAvgPerColSet), 7
- rowAvgPerColSet, matrix-method (rowAvgPerColSet), 7
- rowAvgPerColSet, numeric-method (rowAvgPerColSet), 7
- rowCollapse, 8, 9
- rowCollapse, array-method (rowCollapse), 8
- rowCollapse, matrix-method (rowCollapse), 8
- rowCollapse, numeric-method (rowCollapse), 8
- rowCounts, 10, 11
- rowCounts, array-method (rowCounts), 10
- rowCounts, matrix-method (rowCounts), 10
- rowCounts, numeric-method (rowCounts), 10
- rowCummaxs, 11, 12
- rowCummaxs, array-method (rowCummaxs), 11
- rowCummaxs, matrix-method (rowCummaxs), 11
- rowCummaxs, numeric-method (rowCummaxs), 11
- rowCummins, 13, 14
- rowCummins, array-method (rowCummins), 13
- rowCummins, matrix-method (rowCummins), 13
- rowCummins, numeric-method (rowCummins), 13
- rowCumprods, 14, 15
- rowCumprods, array-method (rowCumprods), 14
- rowCumprods, matrix-method (rowCumprods), 14
- rowCumprods, numeric-method (rowCumprods), 14
- rowCumsums, 16, 17
- rowCumsums, array-method (rowCumsums), 16
- rowCumsums, matrix-method (rowCumsums), 16
- rowCumsums, numeric-method (rowCumsums), 16
- rowDiffs, 17, 18
- rowDiffs, array-method (rowDiffs), 17
- rowDiffs, matrix-method (rowDiffs), 17
- rowDiffs, numeric-method (rowDiffs), 17
- rowIQRDiffs, 19, 20
- rowIQRDiffs, array-method (rowIQRDiffs), 19
- rowIQRDiffs, matrix-method (rowIQRDiffs), 19
- rowIQRDiffs, numeric-method (rowIQRDiffs), 19
- rowIQRs, 20, 20, 21
- rowIQRs, array-method (rowIQRs), 20
- rowIQRs, matrix-method (rowIQRs), 20
- rowIQRs, numeric-method (rowIQRs), 20
- rowLogSumExps, 21, 22
- rowLogSumExps, array-method (rowLogSumExps), 21
- rowLogSumExps, matrix-method (rowLogSumExps), 21
- rowLogSumExps, numeric-method (rowLogSumExps), 21
- rowMadDiffs, 23, 24
- rowMadDiffs, array-method (rowMadDiffs), 23
- rowMadDiffs, matrix-method (rowMadDiffs), 23
- rowMadDiffs, numeric-method (rowMadDiffs), 23
- rowMads, 24, 26, 50
- rowMads(), 21
- rowMads, array-method (rowMads), 24
- rowMads, matrix-method (rowMads), 24
- rowMads, numeric-method (rowMads), 24
- rowMaxs, 12, 26, 27, 31, 37
- rowMaxs, array-method (rowMaxs), 26
- rowMaxs, matrix-method (rowMaxs), 26
- rowMaxs, numeric-method (rowMaxs), 26
- rowMeans, 26, 28, 30, 42, 44, 48

- rowMeans2, [26](#), [27](#), [28](#), [30](#), [42](#), [44](#), [48](#), [51](#)
- rowMeans2, array-method (rowMeans2), [27](#)
- rowMeans2, matrix-method (rowMeans2), [27](#)
- rowMeans2, numeric-method (rowMeans2), [27](#)
- rowMedians, [29](#), [30](#), [53](#)
- rowMedians, array-method (rowMedians), [29](#)
- rowMedians, matrix-method (rowMedians), [29](#)
- rowMedians, numeric-method (rowMedians), [29](#)
- rowMins, [14](#), [27](#), [30](#), [31](#), [37](#)
- rowMins, array-method (rowMins), [30](#)
- rowMins, matrix-method (rowMins), [30](#)
- rowMins, numeric-method (rowMins), [30](#)
- rowOrderStats, [32](#), [33](#)
- rowOrderStats, array-method (rowOrderStats), [32](#)
- rowOrderStats, matrix-method (rowOrderStats), [32](#)
- rowOrderStats, numeric-method (rowOrderStats), [32](#)
- rowProds, [33](#), [34](#)
- rowProds, array-method (rowProds), [33](#)
- rowProds, matrix-method (rowProds), [33](#)
- rowProds, numeric-method (rowProds), [33](#)
- rowQuantiles, [34](#), [36](#)
- rowQuantiles, array-method (rowQuantiles), [34](#)
- rowQuantiles, matrix-method (rowQuantiles), [34](#)
- rowQuantiles, numeric-method (rowQuantiles), [34](#)
- rowRanges, [36](#), [37](#)
- rowRanges, array-method (rowRanges), [36](#)
- rowRanges, matrix-method (rowRanges), [36](#)
- rowRanges, numeric-method (rowRanges), [36](#)
- rowRanks, [38](#), [39](#)
- rowRanks, array-method (rowRanks), [38](#)
- rowRanks, matrix-method (rowRanks), [38](#)
- rowRanks, numeric-method (rowRanks), [38](#)
- rowSdDiffs, [39](#), [41](#)
- rowSdDiffs, array-method (rowSdDiffs), [39](#)
- rowSdDiffs, matrix-method (rowSdDiffs), [39](#)
- rowSdDiffs, numeric-method (rowSdDiffs), [39](#)
- rowSds, [21](#), [26](#), [41](#), [42](#), [48](#), [54](#)
- rowSds(), [41](#)
- rowSds, array-method (rowSds), [41](#)
- rowSds, matrix-method (rowSds), [41](#)
- rowSds, numeric-method (rowSds), [41](#)
- rowSums2, [34](#), [43](#), [44](#)
- rowSums2(), [22](#)
- rowSums2, array-method (rowSums2), [43](#)
- rowSums2, matrix-method (rowSums2), [43](#)
- rowSums2, numeric-method (rowSums2), [43](#)
- rowTabulates, [44](#), [45](#)
- rowTabulates, array-method (rowTabulates), [44](#)
- rowTabulates, matrix-method (rowTabulates), [44](#)
- rowTabulates, numeric-method (rowTabulates), [44](#)
- rowVarDiffs, [45](#), [47](#)
- rowVarDiffs, array-method (rowVarDiffs), [45](#)
- rowVarDiffs, matrix-method (rowVarDiffs), [45](#)
- rowVarDiffs, numeric-method (rowVarDiffs), [45](#)
- rowVars, [28](#), [42](#), [47](#), [48](#), [55](#)
- rowVars(), [47](#)
- rowVars, array-method (rowVars), [47](#)
- rowVars, matrix-method (rowVars), [47](#)
- rowVars, numeric-method (rowVars), [47](#)
- rowWeightedMads, [49](#), [50](#)
- rowWeightedMads, array-method (rowWeightedMads), [49](#)
- rowWeightedMads, matrix-method (rowWeightedMads), [49](#)
- rowWeightedMads, numeric-method (rowWeightedMads), [49](#)
- rowWeightedMeans, [50](#), [51](#)
- rowWeightedMeans, array-method (rowWeightedMeans), [50](#)
- rowWeightedMeans, matrix-method (rowWeightedMeans), [50](#)
- rowWeightedMeans, numeric-method (rowWeightedMeans), [50](#)
- rowWeightedMedians, [52](#), [53](#)
- rowWeightedMedians, array-method (rowWeightedMedians), [52](#)
- rowWeightedMedians, matrix-method (rowWeightedMedians), [52](#)
- rowWeightedMedians, numeric-method (rowWeightedMedians), [52](#)
- rowWeightedSds, [53](#), [54](#)
- rowWeightedSds, array-method (rowWeightedSds), [53](#)
- rowWeightedSds, matrix-method (rowWeightedSds), [53](#)
- rowWeightedSds, numeric-method (rowWeightedSds), [53](#)
- rowWeightedVars, [54](#), [55](#)

rowWeightedVars,array-method  
    (rowWeightedVars), 54  
rowWeightedVars,matrix-method  
    (rowWeightedVars), 54  
rowWeightedVars,numeric-method  
    (rowWeightedVars), 54

stats::quantile, 36  
sum, 44

table, 45  
TRUE, 3, 6, 11, 19, 21, 22, 24, 25, 27–29, 31,  
    34, 35, 37, 40, 42, 43, 46, 48, 49, 51,  
    52, 54, 55

var, 48  
vector, 3, 5, 6, 8, 9, 11–16, 18–22, 24, 25,  
    27–32, 34, 35, 37–40, 42, 43, 45, 46,  
    48–52, 54, 55