

# Package: dibble (via r-universe)

August 24, 2024

**Type** Package

**Title** Dimensional Data Frames

**Version** 0.3.0.9000

**Description** Provides a 'dibble' that implements data cubes (derived from 'dimensional tibble'), and allows broadcasting by dimensional names.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** dplyr, memoise, pillar, purrr (>= 1.0.0), rlang, tibble, tidy, tidyselect, vctrs

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.1

**Suggests** covr, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Depends** R (>= 4.4)

**URL** <https://github.com/UchidaMizuki/dibble>,  
<https://uchidamizuki.github.io/dibble/>

**BugReports** <https://github.com/UchidaMizuki/dibble/issues>

**Repository** <https://uchidamizuki.r-universe.dev>

**RemoteUrl** <https://github.com/uchidamizuki/dibble>

**RemoteRef** HEAD

**RemoteSha** ce45e39e9ce7e6dc9166b857d33e6c55ff5ddd1c

## Contents

apply	2
as_dibble	3
basic-matrices-arrays	4

broadcast . . . . .	5
diag . . . . .	6
dibble . . . . .	7
dibble_by . . . . .	8
extremes . . . . .	8
ifelse . . . . .	9
is_dibble . . . . .	10
nrow-ncol . . . . .	11
row-colnames . . . . .	12

## Index 13

---

apply	<i>Apply functions over array margins</i>
-------	---

---

### Description

Applying a function to margins of a dibble or array, including a matrix.

### Usage

```
apply(X, MARGIN, FUN, ...)

## Default S3 method:
apply(X, MARGIN, FUN, ..., simplify = TRUE)

## S3 method for class 'tbl_ddd'
apply(X, MARGIN, FUN, ...)

## S3 method for class 'ddd_col'
apply(X, MARGIN, FUN, ...)
```

### Arguments

X	A dibble or array, including a matrix.
MARGIN	An integer or character vector giving the subscripts which the function will be applied over.
FUN	A function to be applied.
...	Optional arguments to FUN.
simplify	A logical indicating whether results should be simplified if possible.

### Details

apply() overrides [base::apply\(\)](#) to make it generic. The default method calls the base version.

### Value

A dibble if X is a dibble. See [base::apply\(\)](#) for the return value of the default method.

**See Also**

[base::apply\(\)](#).

**Examples**

```
x <- array(1:24, 2:4,
           list(axis1 = letters[1:2],
                axis2 = letters[1:3],
                axis3 = letters[1:4]))

apply(x, 2:3, sum)
apply(as_dibble(x), 2:3, sum)

apply(x, c("axis2", "axis3"), sum)
apply(as_dibble(x), c("axis2", "axis3"), sum)
```

---

as\_dibble

*Coerce an object to a dibble*

---

**Description**

as\_dibble() turns an object into a dimensional data frame called a dibble.

**Usage**

```
as_dibble(x, ...)

## Default S3 method:
as_dibble(x, ...)

## S3 method for class 'rowwise_df'
as_dibble(x, ...)

## S3 method for class 'grouped_df'
as_dibble(x, ...)

## S3 method for class 'ddf_col'
as_dibble(x, ...)

## S3 method for class 'tbl_ddf'
as_dibble(x, ...)
```

**Arguments**

x	An object.
...	Unused, for extensibility.

**Value**

A dibble.

---

basic-matrices-arrays *Basic matrices and arrays*

---

**Description**

Create basic matrices and arrays.

**Usage**

```
eye(x, ...)  
  
## Default S3 method:  
eye(x, y = x, ...)  
  
## S3 method for class 'matrix'  
eye(x, ...)  
  
## S3 method for class 'ddf_col'  
eye(x, ...)  
  
## S3 method for class 'tbl_ddf'  
eye(x, ...)  
  
ones(x, ...)  
  
## Default S3 method:  
ones(x, y = x, ...)  
  
## S3 method for class 'array'  
ones(x, ...)  
  
## S3 method for class 'ddf_col'  
ones(x, ...)  
  
## S3 method for class 'tbl_ddf'  
ones(x, ...)  
  
zeros(x, ...)  
  
## Default S3 method:  
zeros(x, y = x, ...)  
  
## S3 method for class 'array'
```

```

zeros(x, ...)

## S3 method for class 'ddf_col'
zeros(x, ...)

## S3 method for class 'tbl_ddf'
zeros(x, ...)

```

### Arguments

x	An object.
...	Other arguments passed on to methods.
y	A scalar integer.

### Details

These functions override base functions to make them generic. The default methods call the base versions.

### Value

A dibble if x is a dibble. Otherwise, returns a matrix or an array.

---

broadcast	<i>Broadcast to a new dimension</i>
-----------	-------------------------------------

---

### Description

Broadcasts the dimension of the object to a new dimension.

### Usage

```

broadcast(x, dim_names = NULL, ...)

## Default S3 method:
broadcast(x, dim_names = NULL, ...)

## S3 method for class 'ddf_col'
broadcast(x, dim_names, ...)

## S3 method for class 'tbl_ddf'
broadcast(x, dim_names, ...)

```

### Arguments

x	A dibble, vector, or array.
dim_names	A character vector or list of dimension names.
...	Unused, for extensibility.

**Details**

Operations between dibles are automatically broadcasted, but for safety reasons, warnings are issued. `broadcast()` can suppress the warnings if `dim_names` matches the dimension of `x`.

**Value**

A dibble.

**Examples**

```
x <- broadcast(1:2,
               list(axis1 = letters[1:2]))
y <- broadcast(1:3,
               list(axis2 = letters[1:3]))
broadcast(x * y, c("axis1", "axis2"))
```

---

 diag

*Matrix diagonals*


---

**Description**

Extract or replace the diagonal of a matrix, or construct a diagonal matrix.

**Usage**

```
diag(x, ...)
```

## Default S3 method:  
`diag(x = 1, nrow, ncol, names, ...)`

## S3 method for class 'tbl\_ddf'  
`diag(x, axes, ...)`

## S3 method for class 'ddf\_col'  
`diag(x, axes, ...)`

`diag(x, ...) <- value`

## Default S3 replacement method:  
`diag(x, ...) <- value`

## S3 replacement method for class 'tbl\_ddf'  
`diag(x, ...) <- value`

## S3 replacement method for class 'ddf\_col'  
`diag(x, ...) <- value`

**Arguments**

<code>x</code>	A dibble, matrix, vector or 1D array, or missing.
<code>...</code>	Unused, for extensibility.
<code>nrow, ncol</code>	Optional dimensions for the result when <code>x</code> is not a matrix.
<code>names</code>	(When <code>x</code> is a matrix) logical indicating if the resulting vector, the diagonal of <code>x</code> , should inherit names from <code>dimnames(x)</code> if available.
<code>axes</code>	A character vector of axes.
<code>value</code>	Replacement values.

**Details**

These functions override base functions to make them generic. The default methods call the base versions.

**Value**

A dibble if `x` is a dibble. See `base::diag()` for the return values of the default methods.

---

dibble	<i>Build a dimensional data frame</i>
--------	---------------------------------------

---

**Description**

`dibble()` constructs a dimensional data frame called a dibble.

**Usage**

```
dibble(..., .dim_names = NULL)
```

**Arguments**

<code>...</code>	A set of name-measure pairs.
<code>.dim_names</code>	A list of dimension names.

**Details**

Manipulation functions:

- `mutate()`
- `rename()`
- `select()` & `relocate()`
- `slice()`

**Value**

A dibble.

---

dibble_by	<i>Constructs a dibble by one or more variables</i>
-----------	---

---

**Description**

dibble\_by() constructs a dibble by one or more variables.

**Usage**

```
dibble_by(x, ..., .names_sep = NULL)
```

**Arguments**

x	A data frame or a dibble.
...	Variables.
.names_sep	Passed to tidyr::pack().

**Value**

A dibble.

---

extremes	<i>Maxima and Minima</i>
----------	--------------------------

---

**Description**

Returns the parallel maxima and minima of the input values.

**Usage**

```
pmax(..., na.rm = FALSE)

## Default S3 method:
pmax(..., na.rm = FALSE)

## S3 method for class 'ddf_col'
pmax(..., na.rm = FALSE)

## S3 method for class 'tbl_ddf'
pmax(..., na.rm = FALSE)

pmin(..., na.rm = FALSE)

## Default S3 method:
pmin(..., na.rm = FALSE)
```



```
## S3 method for class 'ddf_col'  
pmin(..., na.rm = FALSE)  
  
## S3 method for class 'tbl_ddf'  
pmin(..., na.rm = FALSE)
```

### Arguments

...                   Dibbles, numeric or character arguments.  
na.rm                 a logical indicating whether missing values should be removed.

### Details

These functions override base functions to make them generic. The default methods call the base versions.

### Value

A dibble if ... are dibles. See [base::pmax\(\)](#) and [base::pmin\(\)](#) for the return value of the default method.

### See Also

[base::pmax\(\)](#), [base::pmin\(\)](#).

---

ifelse

*Conditional element selection*

---

### Description

Selects elements from either yes or no depending on whether test is TRUE or FALSE.

### Usage

```
ifelse(test, yes, no, ...)  
  
## Default S3 method:  
ifelse(test, yes, no, ...)  
  
## S3 method for class 'tbl_ddf'  
ifelse(test, yes, no, ...)  
  
## S3 method for class 'ddf_col'  
ifelse(test, yes, no, ...)
```

**Arguments**

test	An object which can be coerced to logical mode.
yes	Return values for true elements of test.
no	Return values for false elements of test.
...	Unused, for extensibility.

**Details**

ifelse() overrides `base::ifelse()` to make it generic. The default method calls the base version.

**Value**

A dibble if test is a dibble. See `base::ifelse()` for the return value of the default method.

**See Also**

`base::ifelse()`.

---

is_dibble	<i>Test if the object is a dibble</i>
-----------	---------------------------------------

---

**Description**

Test if the object is a dibble

**Usage**

```
is_dibble(x)
```

**Arguments**

x	An object.
---	------------

**Value**

A logical.

---

nrow-ncol	<i>The number of rows/columns</i>
-----------	-----------------------------------

---

**Description**

nrow() and ncol() return the number of rows or columns present in x.

**Usage**

```
nrow(x, ...)  
  
## Default S3 method:  
nrow(x, ...)  
  
## S3 method for class 'ddf_col'  
nrow(x, ...)  
  
## S3 method for class 'tbl_ddf'  
nrow(x, ...)  
  
ncol(x, ...)  
  
## Default S3 method:  
ncol(x, ...)  
  
## S3 method for class 'ddf_col'  
ncol(x, ...)  
  
## S3 method for class 'tbl_ddf'  
ncol(x, ...)
```

**Arguments**

x	An object.
...	Other arguments passed on to methods.

**Details**

These functions override base functions to make them generic. The default methods call the base versions.

**Value**

An integer or NULL.

---

row-colnames	<i>Row and column names</i>
--------------	-----------------------------

---

**Description**

Retrieve or set the row or column names of a matrix-like object.

**Usage**

```
rownames(x, ...)  
  
## Default S3 method:  
rownames(x, ...)  
  
## S3 method for class 'ddf_col'  
rownames(x, ...)  
  
## S3 method for class 'tbl_ddf'  
rownames(x, ...)  
  
colnames(x, ...)  
  
## Default S3 method:  
colnames(x, ...)  
  
## S3 method for class 'ddf_col'  
colnames(x, ...)  
  
## S3 method for class 'tbl_ddf'  
colnames(x, ...)
```

**Arguments**

x	A matrix-like object.
...	Other arguments passed on to methods.

**Details**

These functions override base functions to make them generic. The default methods call the base versions.

**Value**

A list of row/column names.

# Index

`apply`, 2  
`as_dibble`, 3

`base::apply()`, 2, 3  
`base::diag()`, 7  
`base::ifelse()`, 10  
`base::pmax()`, 9  
`base::pmin()`, 9  
`basic-matrices-arrays`, 4  
`broadcast`, 5

`colnames (row-colnames)`, 12

`diag`, 6  
`diag<- (diag)`, 6  
`dibble`, 7  
`dibble_by`, 8

`extremes`, 8  
`eye (basic-matrices-arrays)`, 4

`ifelse`, 9  
`is_dibble`, 10

`ncol (nrow-ncol)`, 11  
`nrow (nrow-ncol)`, 11  
`nrow-ncol`, 11  
`nrow.ddf_col (nrow-ncol)`, 11  
`nrow.default (nrow-ncol)`, 11  
`nrow.tbl_ddf (nrow-ncol)`, 11

`ones (basic-matrices-arrays)`, 4

`pmax (extremes)`, 8  
`pmin (extremes)`, 8

`row-colnames`, 12  
`rownames (row-colnames)`, 12

`zeros (basic-matrices-arrays)`, 4