# Package ‘broadcast’

December 31, 2024

**Title** Simple Operations Broadcasted for Atomic Arrays and Multi-Threaded for Atomic Long Vectors

**Version** 0.0.0.9000

**Description** Implements simple broadcasted operations for atomic arrays,
and simple parallel computed operations for atomic long vectors.
Includes relational operators (`==`, `!=`, `<`, `>`, `<=`, `>=`),
boolean combiner operations (`&`, `|`, xor(), ``not-and''),
integer/double/complex arithmetic operators (`+`, `-`, `*`, `/`, `^` `%%`, pmin(), pmax()),
and string concatination operators(`+` , `*`).
The broadcasted operators have about the same performance as their base 'R' counterparts.
The broadcasted operators use considerably less memory than their base 'R' counterparts.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LinkingTo** Rcpp

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.1

**Depends** R (>= 4.2.0)

**Imports** Rcpp (>= 1.0.11)

**Suggests** tinytest

## Contents

## `aaa00_broadcast_help` *broadcast: Subset Methods as Alternatives to the Square Brackets Operators for Programming*

### Description

broadcast:
Simple broadcasted infix operations for atomic arrays in 'R'.

The 'broadcast' package supports relational operators (==, !=, <, > <=, >=), logical combiners (&, |, xor, nand), arithmetic (+, -, *, /, ^)

### Author(s)

**Author, Maintainer**: Tony Wilkes `<tony_a_wilkes@outlook.com>` (ORCID)

### References

The badges shown in the documentation of this R-package were made using the services of: `https://shields.io/`

## `array_recycle` *Recycle Array Dimensions*

### Description

The `array_recycle()` function recycles array dimensions until the specified dimension sizes are reached, and returns the array.

The various broadcasting functions "recycle" an array virtually, meaning little to no additional memory is needed.
The `array_recycle()` function, however, physically recycles an array (and thus actually occupies memory space).

### Usage

```
array_recycle(x, tdim)
```

### Arguments

| | |
|---|---|
| `x` | an atomic or recursive array or matrix. |
| `tdim` | an integer vector, giving the target dimension to reach. |

### Value

Returns the recycled array.

### Examples

```
x <- matrix(1:9, 3,3)
colnames(x) <- LETTERS[1:3]
rownames(x) <- letters[1:3]
names(x) <- month.abb[1:9]
print(x)

array_recycle(x, c(3,3,2)) # recycle to larger size
```

---

## `atomic_typecast` *Atomic Type Casting With Names and Dimensions Preserved*

---

### Description

Atomic type casting in R is generally performed using the functions as.logical, as.integer, as.double, as.character, as.complex, and as.raw.

Converting an object between atomic types using these functions strips the object of its attributes, including (dim)names and dimensions.

The functions provided here by the 'tinycodet' package preserve the dimensions, dimnames, and names.

The functions are as follows:

- `as_bool()`: converts object to atomic type `logical` (`TRUE, FALSE, NA`).
- `as_int()`: converts object to atomic type `integer`.
- `as_dbl()`: converts object to atomic type `double` (AKA decimal numbers).
- `as_chr()`: converts object to atomic type `character`.
- `as_cplx()`: converts object to atomic type `complex`.
- `as_raw()`:converts object to atomic type `raw`.

### Usage

```
as_bool(x, ...)

as_int(x, ...)

as_dbl(x, ...)

as_chr(x, ...)

as_cplx(x, ...)

as_raw(x, ...)
```

### Arguments

| | |
|---|---|
| `x` | vector, matrix, array (or a similar object where all elements share the same type). |
| `...` | further arguments passed to or from other methods. |

## Value

The converted object.

## Examples

```
# matrix example ====
x <- matrix(sample(-1:28), ncol = 5)
colnames(x) <- month.name[1:5]
rownames(x) <- month.abb[1:6]
names(x) <- c(letters[1:20], LETTERS[1:10])
print(x)

as_bool(x)
as_int(x)
as_dbl(x)
as_chr(x)
as_cplx(x)
as_raw(x)


################################################################################

# factor example ====
x <- factor(month.abb, levels = month.abb)
names(x) <- month.name
print(x)

as_bool(as_int(x) > 6)
as_int(x)
as_dbl(x)
as_chr(x)
as_cplx(x)
as_raw(x)
```

---

## bc.d *Broadcasted Decimal Arithmetic*

---

## Description

The `bc.d()` function performs broadcasted decimal arithmetic operations on 2 atomic arrays.

## Usage

```
bc.d(x, y, op)
```

## Arguments

| | |
|---|---|
| x, y | conformable atomic arrays of types `logical`, `integer`, or `double`. |
| op | a single string, giving the operator.<br>Supported operators: +, -, *, /, ^. |

## Value

The list.

## Examples

```
x.dim <- c(10:8)
x.len <- prod(x.dim)
x.data <- sample(c(NA, 1.1:1000.1), x.len, TRUE)
x <- array(x.data, x.dim)
y <- array(1:50, c(10,1,1))

bc.d(x, y, "+")
bc.d(x, y, "-")
bc.d(x, y, "*")
bc.d(x, y, "/")
bc.d(x, y, "^")
```

---

`bc_pred_dim` *Predict Broadcasted dimensions*

---

## Description

`bc_pred_dim()`

## Usage

```
bc_pred_dim(x, y)
```

## Arguments

| | |
|---|---|
| `x, y` | an atomic array or matrix. |

## Value

Returns the recycled array.

## Examples

```
x <- matrix(1:9, 3,3)
colnames(x) <- LETTERS[1:3]
rownames(x) <- letters[1:3]
names(x) <- month.abb[1:9]
print(x)

array_recycle(x, c(3,3,2)) # recycle to larger size
```

# Index