

# Package ‘pipebind’

August 30, 2023

**Type** Package

**Title** Flexible Binding for Complex Function Evaluation with the Base R  
> Pipe

**Version** 0.1.2

**Maintainer** Brenton M. Wiernik <brenton@wiernik.org>

**Description** Provides a simple function to bind a piped object to a placeholder  
symbol to enable complex function evaluation with the base R > pipe.

**License** GPL-3

**Encoding** UTF-8

**Language** en-US

**URL** <https://github.com/bwiernik/pipebind/>

**BugReports** <https://github.com/bwiernik/pipebind/issues>

**RoxygenNote** 7.2.3.9000

**NeedsCompilation** no

**Author** Brenton M. Wiernik [cre, aut] (<<https://orcid.org/0000-0001-9560-6336>>,  
@bmiernik)

**Repository** CRAN

**Date/Publication** 2023-08-30 18:40:02 UTC

## R topics documented:

bind . . . . .	2
bracket . . . . .	3
<b>Index</b>	<b>5</b>

---

 bind

*Bind a (piped) object to a symbol for complex function evaluation*


---

### Description

The base R `|>` pipe lacks some advanced functionality compared to the `{magrittr}%>%` pipe. For example, the piped object can only appear once on the right-hand side of the pipe (either as the first unnamed argument or elsewhere using the `_` placeholder in R 4.2.0 and later), and the `_` placeholder cannot appear on the left side of sub-setting functions like `$`, `[`, `[[`, or `@`.

The `bind()` function is a way to conveniently circumvent these limitations. Pipe an object into `bind()`, choose a placeholder symbol to represent it, then use this placeholder to refer the piped object in any way and as many times as desired in an R expression.

The Greek letter  $\lambda()$  is available as an alias for `bind()`.

### Usage

```
bind(.pipeValue, .pipeBind, ...)
```

### Arguments

<code>.pipeValue</code>	The object to bind. Typically specified by piping into the <code>bind()</code> function (e.g., <code>x  &gt; bind()</code> ).
<code>.pipeBind</code>	The placeholder symbol to use to represent the piped object. Can be any valid R object name.
<code>...</code>	An R expression. Any valid R code (expression).

### Value

The results of the expression, evaluated using the piped object.

### Examples

```
# Piping to a non-first argument
mtcars |>
  transform(kmL = mpg / 2.35) |>
  bind(d, lm(kmL ~ hp, data = d))

# Using the piped value multiple times
rnorm(10, mean = 10) |>
  bind(x, x - mean(x))

# Using the piped value in multiple arguments
c(a = 1, b = 2, c = 3) |>
  bind(x, paste(names(x), x, sep = " = "))

# Subsetting the piped value
mtcars |>
  bind(d, d$mpg)
```

bracket

*Pipe-able aliases***Description**

pipebind provides several aliases for unary/binary operators (e.g., +) and replacement functions (e.g., names<-()) that facilitate using these functions in a |> chain.

Some unary/binary operators cannot currently be used with the |> pipe, such as +, -, or %\*%. These aliases provide a way to use these functions with the |> pipe.

Currently implemented aliases are

**Extract and replace elements**

bracket	'['
double_bracket	'[['
assign_bracket	'[<-'
assign_double_bracket	'[[<-'
dollar	'\$'
at_sign	'@'

**Arithmetic operators**

add	'+'
subtract	'-'
multiply	'*'
divide	'/'
integer_divide	'%/%'
mod	'%%'
raise_to_power	'^'
matrix_multiply	'%*%'

**Logical comparisons**

and	'&'
or	' '
not	'!'
single_and	'&&'
single_or	'  '
equals	'=='
greater_than	'>'
greater_or_equal	'>='
less_than	'<'
less_or_equal	'<='
is_in	'%in%'

**Assign attributes**

assign_names	'names<-'
assign_colnames	'colnames<-'
assign_rownames	'rownames<-'
assign_dimnames	'dimnames<-'
assign_class	'class<-'
assign_attributes	'attributes<-'

assign_attr	'attr<-'
assign_levels	'levels<-'
assign_contrasts	'contrasts<-'
assign_units	'units<-'
assign_comment	'comment<-'
assign_diag	'diag<-'
assign_dim	'dim<-'
assign_length	'length<-'
assign_as_na	'is.na<-'

### Note

Inspired and some alias names adapted from from *magrittr*. Reused code Copyright (c) 2023 magrittr authors.

### Examples

```
mtcars |>
  bracket(, 1:4)

1:10 |>
  add(5) |>
  matrix(dimnames = list(letters[1:10], "x")) |>
  matrix_multiply(seq(10, 100, by = 10))

data.frame(1:10, letters[1:10]) |>
  assign_names(c("numbers", "letters"))
```

# Index

add (bracket), 3  
and (bracket), 3  
assign\_as\_na (bracket), 3  
assign\_attr (bracket), 3  
assign\_attributes (bracket), 3  
assign\_bracket (bracket), 3  
assign\_class (bracket), 3  
assign\_colnames (bracket), 3  
assign\_comment (bracket), 3  
assign\_contrasts (bracket), 3  
assign\_diag (bracket), 3  
assign\_dim (bracket), 3  
assign\_dimnames (bracket), 3  
assign\_double\_bracket (bracket), 3  
assign\_length (bracket), 3  
assign\_levels (bracket), 3  
assign\_names (bracket), 3  
assign\_rownames (bracket), 3  
assign\_units (bracket), 3  
at\_sign (bracket), 3

base R, 2  
bind, 2  
bracket, 3

divide (bracket), 3  
dollar (bracket), 3  
double\_bracket (bracket), 3

equals (bracket), 3

greater\_or\_equal (bracket), 3  
greater\_than (bracket), 3

integer\_divide (bracket), 3  
is\_in (bracket), 3

lambda (bind), 2  
less\_or\_equal (bracket), 3  
less\_than (bracket), 3

matrix\_multiply (bracket), 3  
mod (bracket), 3  
multiply (bracket), 3

not (bracket), 3

or (bracket), 3

pipe, 2  
pipebind (bind), 2

raise\_to\_power (bracket), 3

single\_and (bracket), 3  
single\_or (bracket), 3  
subtract (bracket), 3