

# Package: pipebind (via r-universe)

August 24, 2024

**Type** Package

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

**Version** 0.1.2.0001

**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

**LazyData** true

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

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

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3.9000

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

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

**RemoteRef** HEAD

**RemoteSha** a6e454bbd79ccbf24cdf1d3ca52e4fd869c6ac4b

## Contents

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