

# Package ‘magrittr’

July 2, 2014

**Type** Package

**Title** magrittr - a forward-pipe operator for R

**Version** 1.0.1

**Date** 2014-05-14

**Author** Stefan Milton Bache <stefan@stefanbache.dk> and  
Hadley Wickham <h.wickham@gmail.com>

**Maintainer** Stefan Milton Bache <stefan@stefanbache.dk>

**Description** Provides a mechanism for chaining commands with a  
new forward-pipe operator. Ceci n'est pas un pipe.

**Suggests** testthat, knitr

**VignetteBuilder** knitr

**License** MIT + file LICENSE

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-05-15 21:12:27

## R topics documented:

debug_pipe . . . . .	2
extract . . . . .	2
magrittr . . . . .	3
%>% . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

debug_pipe	<i>Place a debug browser in a chain of commands.</i>
------------	--

---

### Description

When using magrittr's piping syntax, it can be useful to to debug at certain steps within a chain. This is a simple wrapper around browser for this purpose.

### Usage

```
debug_pipe(x)
```

### Arguments

x	A value.
---	----------

### Value

returns the argument x.

### Examples

```
iris %>%
  debug_pipe %>%
  head
```

---

extract	<i>Aliases</i>
---------	----------------

---

### Description

magrittr provides a series of aliases which can be more pleasant to use when composing chains using the %>% operator.

### Details

Currently implemented aliases are

extract	'[ '
extract2	'[[ '
use_series	'\$ '
add	'+'
subtract	'- '
multiply_by	'* '
raise_to_power	'^ '
multiply_by_matrix	'%*%'

```

divide_by           '/'
divide_by_int       '%/%'
mod                 '%%'
and                 '&'
or                  '|'
equals              '=='
is_greater_than     '>'
is_weakly_greater_than '>='
is_less_than        '<'
is_weakly_less_than '<='
not                 '!'
set_colnames        'colnames<- '
set_rownames        'rownames<- '
set_names           'names<- '

```

## Examples

```

iris %>%
  extract(, 1:4) %>%
  head

good.times <-
  Sys.Date() %>%
  as.POSIXct %>%
  seq(by = "15 mins", length.out = 100) %>%
  data.frame(timestamp = .)

good.times$quarter <-
  good.times %>%
  use_series(timestamp) %>%
  format("%M") %>%
  as.numeric %>%
  divide_by_int(15) %>%
  add(1)

```

---

magrittr

*magrittr - a forward-pipe operator for R*

---

## Description

This package provides pipe-forwarding mechanisms similar to (but not exactly like) e.g. F#'s pipe-forward operator. The semantics of magrittr yield clean and readable code which can be read from left to right, rather than from the inside and out (which is the case with nested function calls), and reduce the need for temporary value bindings. This is particularly useful for manipulating data frames etc, where multiple steps are needed before arriving at the desired result. The package also contains useful functions/aliases which fit well into the syntax promoted by the package. For a brief introduction, see `vignette("magrittr")`.

**Details**

Package: magrittr  
 Type: Package  
 Version: 1.0.1  
 Date: 2014-01-14  
 License: MIT + file LICENCE.

The main feature is provided by the %>% binary operator. It takes a value (e.g. a data.frame) on the left-hand side and a function or expression on the right-hand side, see %>%.

**Author(s)**

Stefan Milton Bache and Hadley Wickham.

Maintainer: Stefan Holst Milton Bache <stefan[at]stefanbache.dk>

---

%>%

*Pipe an object forward into a function call/expression.*

---

**Description**

The %>% operator pipes the left-hand side into an expression on the right-hand side. The expression can contain a . as placeholder to indicate the position taken by the object in the pipeline. If not present, value will be placed as the first argument. If the right-hand side expression is a function call that takes only one argument, one can omit parentheses and the .. Only the outmost call is matched against the dot, which means that e.g. formulas can still use a dot which will not be matched. Nested functions will not be matched either.

**Usage**

lhs %>% rhs

**Arguments**

lhs	The value to be piped
rhs	A function or expression

**Details**

When the right-hand side expression is enclosed in parentheses, it is evaluated before the left-hand side is passed on, which can be used when the right-hand side itself evaluates to the relevant function or call. It is advised, but not strictly necessary, to parenthesize anonymous function definitions when used in pipelines.

**Examples**

```
## Not run:
library(dplyr)
library(Lahman)

Batting %>%
  group_by(playerID) %>%
  summarise(total = sum(G)) %>%
  arrange(desc(total)) %>%
  head(5)

iris %>%
  filter(Petal.Length > 5) %>%
  select(-Species) %>%
  colMeans

iris %>%
  aggregate(. ~ Species, ., mean)

rnorm(1000) %>% abs %>% sum

## End(Not run)

rnorm(1000) %>%
  (function(x) {
    hist(x)
    x
  })

1:10 %>% (call("sum"))

1:10 %>% (substitute(a(), list(a = sum)))
```

# Index

`%>%`, [4](#), [4](#)

`add` (extract), [2](#)

`and` (extract), [2](#)

`debug_pipe`, [2](#)

`divide_by` (extract), [2](#)

`divide_by_int` (extract), [2](#)

`equals` (extract), [2](#)

`extract`, [2](#)

`extract2` (extract), [2](#)

`is_greater_than` (extract), [2](#)

`is_less_than` (extract), [2](#)

`is_weakly_greater_than` (extract), [2](#)

`is_weakly_less_than` (extract), [2](#)

`magrittr`, [3](#)

`magrittr-package` (magrittr), [3](#)

`mod` (extract), [2](#)

`multiply_by` (extract), [2](#)

`multiply_by_matrix` (extract), [2](#)

`not` (extract), [2](#)

`or` (extract), [2](#)

`raise_to_power` (extract), [2](#)

`set_colnames` (extract), [2](#)

`set_names` (extract), [2](#)

`set_rownames` (extract), [2](#)

`subtract` (extract), [2](#)

`use_series` (extract), [2](#)