

# Package ‘dcmodyfy’

July 22, 2025

**Maintainer** Mark van der Loo <mark.vanderloo@gmail.com>

**License** GPL-3

**Title** Modify Data Using Externally Defined Modification Rules

**Type** Package

**LazyLoad** yes

**Description** Data cleaning scripts typically contain a lot of 'if this change that' type of statements. Such statements are typically condensed expert knowledge. With this package, such 'data modifying rules' are taken out of the code and become in stead parameters to the work flow. This allows one to maintain, document, and reason about data modification rules as separate entities.

**Version** 0.9.0

**Depends** methods

**URL** <https://github.com/data-cleaning/dcmodyfy>

**BugReports** <https://github.com/data-cleaning/dcmodyfy/issues>

**Encoding** UTF-8

**Imports** yaml, validate (>= 1.1.3), lumberjack (>= 1.3.1), settings, utils,

**Suggests** simplermardown, tinytest,

**VignetteBuilder** simplermardown

**Collate** 'dplyr\_verbs.R' 'guard.R' 'modifier.R' 'modify.R' 'validate.R'

**RoxygenNote** 7.3.1

**NeedsCompilation** no

**Author** Mark van der Loo [cre, aut] (ORCID:

<<https://orcid.org/0000-0002-9807-4686>>),

Edwin de Jonge [aut] (ORCID: <<https://orcid.org/0000-0002-6580-4718>>),

Sjabbo Schaveling [ctb],

Floris Ruijter [ctb]

**Repository** CRAN

**Date/Publication** 2024-03-28 08:20:06 UTC

## Contents

dcmmodify . . . . .	2
modifier . . . . .	3
modify . . . . .	3
modify_so . . . . .	4

<b>Index</b>	<b>5</b>
--------------	----------

---

dcmmodify	<i>Data Modification By Modifying Rules</i>
-----------	---

---

## Description

Data often contain errors and missing data. Experts can often correct commonly occurring errors based on simple conditional rules. This package facilitates the expression, management, and application of such rules on data sets.

The general workflow in dcmmodify follows the following pattern.

- Define or read a set of rules with `modifier`.
- `modify` data with the modification rules.
- Examine the results either graphically or by summary.

There are several convenience functions that allow one to define modification rules from the commandline, through a (freeform or yaml) file and to investigate and maintain the rules themselves. Please have a look at the introductory vignette

```
vignette("introduction", package="dcmmodify")
```

## Author(s)

**Maintainer:** Mark van der Loo <mark.vanderloo@gmail.com> ([ORCID](#))

Authors:

- Edwin de Jonge ([ORCID](#))

Other contributors:

- Sjabbo Schaveling [contributor]
- Floris Ruijter [contributor]

## See Also

Useful links:

- <https://github.com/data-cleaning/dcmmodify>
- Report bugs at <https://github.com/data-cleaning/dcmmodify/issues>

---

modifier	<i>Create or read a set of data modification rules</i>
----------	--

---

**Description**

Create or read a set of data modification rules

**Usage**

```
modifier(..., .file, .data)
```

**Arguments**

...	A comma-separated list of modification rules.
.file	(optional) A character vector of file locations.
.data	(optional) A data.frame with at least a column "rule" of type character. Optionally, the following columns of metadata can be provided (all character, except "created" which should be POSIXct): "name", "label", "description", "origin", "created".

**Value**

An object of class modifier.

**Examples**

```
m <- modifier( if (height < mean(height)) height <- 2*height
, if ( weight > mean(weight) ) weight <- weight/2 )
modify(women,m)
```

---

modify	<i>Modify a data set</i>
--------	--------------------------

---

**Description**

Modify a data set

**Usage**

```

modify(dat, x, ref, ...)

## S4 method for signature 'data.frame,modifier,environment'
modify(dat, x, ref, logger = NULL, ...)

## S4 method for signature 'data.frame,modifier,ANY'
modify(dat, x, logger = NULL, ...)

## S4 method for signature 'data.frame,modifier,data.frame'
modify(dat, x, ref, logger = NULL, ...)

## S4 method for signature 'data.frame,modifier,list'
modify(dat, x, ref, logger = NULL, ...)

```

**Arguments**

dat	A data.frame
x	A modifier object containing modifying rules.
ref	A environment
...	Extra arguments.
logger	Optional. A lumberjack-compatible logger object.

**Examples**

```

m <- modifier( if (height < mean(height)) height <- 2*height
, if ( weight > mean(weight) ) weight <- weight/2 )
modify(women,m)

```

---

modify_so	<i>Shortcut to modify data</i>
-----------	--------------------------------

---

**Description**

Shortcut to modify data

**Usage**

```

modify_so(dat, ...)

```

**Arguments**

dat	A data.frame
...	A comma-separated list of modifying rules.

# Index

dcmodify, 2  
dcmodify-package (dcmodify), 2

modifier, 2, 3  
modify, 2, 3  
modify, data.frame, modifier, ANY-method  
    (modify), 3  
modify, data.frame, modifier, data.frame-method  
    (modify), 3  
modify, data.frame, modifier, environment-method  
    (modify), 3  
modify, data.frame, modifier, list-method  
    (modify), 3  
modify\_so, 4