

# Package ‘dfidx’

June 8, 2025

**Version** 0.1-2

**Date** 2025-06-06

**Title** Indexed Data Frames

**Depends** R (>= 4.1.0)

**Imports** Formula, Rdpack

**Suggests** knitr, quarto

**Description** Provides extended data frames, with a special data frame column which contains two indexes, with potentially a nesting structure.

**License** GPL (>= 2)

**URL** <https://cran.r-project.org/package=dfidx>

**VignetteBuilder** quarto

**RoxygenNote** 7.3.1

**Encoding** UTF-8

**LazyData** true

**RdMacros** Rdpack

**NeedsCompilation** no

**Author** Yves Croissant [aut, cre]

**Maintainer** Yves Croissant <yves.croissant@univ-reunion.fr>

**Repository** CRAN

**Date/Publication** 2025-06-08 13:40:02 UTC

## Contents

|                             |    |
|-----------------------------|----|
| dfidx . . . . .             | 2  |
| idx . . . . .               | 4  |
| idx_name . . . . .          | 5  |
| methods.dfidx . . . . .     | 6  |
| model.frame.dfidx . . . . . | 8  |
| munnell . . . . .           | 9  |
| unfold_idx . . . . .        | 10 |

**Index****11**


---

|       |                                 |
|-------|---------------------------------|
| dfidx | <i>Data frames with indexes</i> |
|-------|---------------------------------|

---

**Description**

data frames for which observations are defined by two (potentially nested) indexes and for which series have thefore a natural tabular representation

**Usage**

```
dfidx(
  data,
  idx = NULL,
  drop.index = TRUE,
  as.factor = NULL,
  pkg = NULL,
  fancy.row.names = FALSE,
  subset = NULL,
  idnames = NULL,
  shape = c("long", "wide"),
  choice = NULL,
  varying = NULL,
  sep = ".",
  opposite = NULL,
  levels = NULL,
  ranked = FALSE,
  name,
  position,
  ...
)
```

**Arguments**

|                 |                                                                                                                |
|-----------------|----------------------------------------------------------------------------------------------------------------|
| data            | a data frame                                                                                                   |
| idx             | an index                                                                                                       |
| drop.index      | if TRUE (the default), remove the index series from the data.frame as stand alone series                       |
| as.factor       | should the indexes be coerced to factors ?                                                                     |
| pkg             | if set, the resulting dfidx object is of class c("dfidx_pkg", "dfidx") which enables to write specific classes |
| fancy.row.names | if TRUE, fancy row names are computed                                                                          |
| subset          | a logical which defines a subset of rows to return                                                             |
| idnames         | the names of the indexes                                                                                       |

|              |                                                                              |
|--------------|------------------------------------------------------------------------------|
| shape        | either wide or long                                                          |
| choice       | the choice                                                                   |
| varying, sep | relevant for data sets in wide format, these arguments are passed to reshape |
| opposite     | return the opposite of the series                                            |
| levels       | the levels for the second index                                              |
| ranked       | a boolean for ranked data                                                    |
| name         | name of the idx column                                                       |
| position     | position of the idx column                                                   |
| ...          | further arguments                                                            |

### Details

Indexes are stored as a `data.frame` column in the resulting `dfidx` object

### Value

an object of class `"dfidx"`

### Author(s)

Yves Croissant

### Examples

```
# the first two columns contain the index
mn <- dfidx(munnell)

# explicitly indicate the two indexes using either a vector or a
# list of two characters
mn <- dfidx(munnell, idx = c("state", "year"))
mn <- dfidx(munnell, idx = list("state", "year"))

# rename one or both indexes
mn <- dfidx(munnell, idnames = c(NA, "period"))

# for balanced data (with observations ordered by the first, then
# by the second index

# use the name of the first index
mn <- dfidx(munnell, idx = "state", idnames = c("state", "year"))

# or an integer equal to the cardinal of the first index
mn <- dfidx(munnell, idx = 48, idnames = c("state", "year"))

# Indicate the values of the second index using the levels argument
mn <- dfidx(munnell, idx = 48, idnames = c("state", "year"),
            levels = 1970:1986)

# Nesting structure for one of the index
```

```
mn <- dfidx(munnell, idx = c(region = "state", president = "year"))

# Data in wide format
mn <- dfidx(munnell_wide, idx = c(region = "state"),
           varying = 3:36, sep = "_", idnames = c(NA, "year"))

# Customize the name and the position of the `idx` column
#dfidx(munnell, position = 3, name = "index")
```

---

|     |                        |
|-----|------------------------|
| idx | <i>Index for dfidx</i> |
|-----|------------------------|

---

## Description

The index of a `dfidx` is a `data.frame` containing the different series which define the two indexes (with possibly a nesting structure). It is stored as a "sticky" `data.frame` column of the `data.frame` and is also inherited by series (of class `'xseries'`) which are extracted from a `dfidx`.

## Usage

```
idx(x, n = NULL, m = NULL)

## S3 method for class 'dfidx'
idx(x, n = NULL, m = NULL)

## S3 method for class 'idx'
idx(x, n = NULL, m = NULL)

## S3 method for class 'xseries'
idx(x, n = NULL, m = NULL)

## S3 method for class 'idx'
format(x, size = 4, ...)
```

## Arguments

|                   |                                                                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x</code>    | a <code>dfidx</code> or a <code>xseries</code>                                                                                                  |
| <code>n, m</code> | <code>n</code> is the index to be extracted (1 or 2), <code>m</code> equal to one to get the index, greater than one to get a nesting variable. |
| <code>size</code> | the number of characters of the indexes for the format method                                                                                   |
| <code>...</code>  | further arguments (for now unused)                                                                                                              |

## Details

`idx` is defined as a generic with a `dfidx` and a `xseries` method.

**Value**

a data.frame containing the indexes or a series if a specific index is selected

**Author(s)**

Yves Croissant

**Examples**

```
mn <- dfidx(munnell, idx = c(region = "state", president = "year"))
idx(mn)
gsp <- mn$gsp
idx(gsp)
# get the first index
idx(mn, 1)
# get the nesting variable of the first index
idx(mn, 1, 2)
```

---

|          |                                     |
|----------|-------------------------------------|
| idx_name | <i>Get the names of the indexes</i> |
|----------|-------------------------------------|

---

**Description**

This function extract the names of the indexes or the name of a specific index

**Usage**

```
idx_name(x, n = 1, m = NULL)

## S3 method for class 'dfidx'
idx_name(x, n = NULL, m = NULL)

## S3 method for class 'idx'
idx_name(x, n = NULL, m = NULL)

## S3 method for class 'xseries'
idx_name(x, n = NULL, m = NULL)
```

**Arguments**

|   |                                                                    |
|---|--------------------------------------------------------------------|
| x | a dfidx, a idx or a xseries object                                 |
| n | the index to be extracted (1 or 2, ignoring the nesting variables) |
| m | if > 1, a nesting variable                                         |

**Value**

if n is NULL, a named integer which gives the position of the idx column in the dfidx object, otherwise, a character of length 1

**Author(s)**

Yves Croissant

**Examples**

```
mn <- dfidx(munnell, idx = c(region = "state", president = "year"))
# get the position of the idx column
idx_name(mn)
# get the name of the first index
idx_name(mn, 1)
# get the name of the second index
idx_name(mn, 2)
# get the name of the nesting variable for the second index
idx_name(mn, 2, 2)
```

methods.dfidx

*Methods for dfidx***Description**

A dfidx is a data.frame with a "sticky" data.frame column which contains the indexes. Specific methods of functions that extract lines and/or columns of a data.frame are provided.

**Usage**

```
## S3 method for class 'dfidx'
x[i, j, drop]

## S3 method for class 'dfidx'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

## S3 method for class 'dfidx'
print(x, ..., n = NULL)

## S3 method for class 'dfidx'
head(x, n = 10L, ...)

## S3 method for class 'dfidx'
x[[y]]

## S3 method for class 'dfidx'
x$y

## S3 replacement method for class 'dfidx'
object$y <- value

## S3 replacement method for class 'dfidx'
```

```

object[[y]] <- value

## S3 method for class 'xseries'
print(x, ..., n = 10L)

## S3 method for class 'idx'
print(x, ..., n = NULL)

## S3 method for class 'dfidx'
mean(x, ...)

```

### Arguments

|                     |                                                                       |
|---------------------|-----------------------------------------------------------------------|
| x, object           | a dfidx object                                                        |
| i                   | the row index                                                         |
| j                   | the column index                                                      |
| drop                | if TRUE a vector is returned if the result is a one column data.frame |
| row.names, optional | arguments of the generic as.data.frame method, not used               |
| ...                 | further arguments                                                     |
| n                   | the number of rows for the print method                               |
| y                   | the name or the position of the series one wishes to extract          |
| value               | the value for the replacement method                                  |

### Value

as.data.frame and mean return a data.frame, [[ and \$ a vector, [ either a dfidx or a vector, \$<- and [[<- modify the values of an existing column or create a new column of a dfidx object, print is called for its side effect

### Author(s)

Yves Croissant

### Examples

```

mn <- dfidx(munnell)
# extract a series (returns as a xseries object)
mn$gsp
# or
mn[["gsp"]]
# extract a subset of series (returns as a dfidx object)
mn[c("gsp", "unemp")]
# extract a subset of rows and columns
mn[mn$unemp > 10, c("utilities", "water")]
# dfidx, idx and xseries have print methods as (like tibbles), a n
# argument
print(mn, n = 3)

```

```
print(idx(mn), n = 3)
print(mn$gsp, n = 3)
# a dfidx object can be coerced to a data.frame
head(as.data.frame(mn))
```

---

|                   |                                             |
|-------------------|---------------------------------------------|
| model.frame.dfidx | <i>model.frame/matrix for dfidx objects</i> |
|-------------------|---------------------------------------------|

---

## Description

Specific model.frame/matrix are provided for dfidx objects. This leads to an unusual order of arguments compared to the usage. Actually, the first two arguments of the model.frame method are a dfidx and a formula and the only main argument of the model.matrix is a dfidx which should be the result of a call to the model.frame method, i.e. it should have a term attribute.

## Usage

```
## S3 method for class 'dfidx'
model.frame(
  formula,
  data = NULL,
  ...,
  lhs = NULL,
  rhs = NULL,
  dot = "previous",
  alt.subset = NULL,
  refllevel = NULL,
  balanced = FALSE
)

## S3 method for class 'dfidx'
model.matrix(object, ..., lhs = NULL, rhs = 1, dot = "separate")

## S3 method for class 'dfidx_matrix'
print(x, ..., n = 10L)
```

## Arguments

|                    |                                                                            |
|--------------------|----------------------------------------------------------------------------|
| formula            | a dfidx                                                                    |
| data               | a formula                                                                  |
| ..., lhs, rhs, dot | see the Formula method                                                     |
| alt.subset         | a subset of levels for the second index                                    |
| refllevel          | a user-defined first level for the second index                            |
| balanced           | a boolean indicating if the resulting data.frame has to be balanced or not |
| object             | a dfidx object                                                             |
| x                  | a model matrix                                                             |
| n                  | the number of lines to print                                               |



**Value**

a `dfidx` object for the `model.frame` method and a matrix for the `model.matrix` method.

**Author(s)**

Yves Croissant

**Examples**

```
mn <- dfidx(munnell)
mf <- model.frame(mn, gsp ~ privatecap | publiccap + utilities | unemp + labor)
model.matrix(mf, rhs = 1)
model.matrix(mf, rhs = 2)
model.matrix(mf, rhs = 1:3)
```

---

munnell

---

*Productivity in the United States*


---

**Description**

a panel data of 48 American States for 17 years, from 1970 to 1986

**Usage**

```
munnell
```

```
munnell_wide
```

**Format**

a tibble containing:

- state: the state
- year: the year
- region: one of the 9 regions of the United States
- president: the name of the president for the given year
- publiccap: public capital stock
- highway: highway and streets
- water: water and sewer facilities
- utilities: othe public building and structures
- privatecap: private capital stock
- gsp: gross state product
- labor: labor input measured by the employment in non-agricultural payrolls
- unemp: state unemployment rate

An object of class `data.frame` with 48 rows and 36 columns.

## Source

online complements to Baltagi (2001): <https://www.wiley.com/legacy/wileychi/baltagi/>  
 Online complements to Baltagi (2013): <https://bcs.wiley.com/he-bcs/Books?action=resource&bcsId=4338&itemId=1118672321&resourceId=13452>

## References

Baltagi BH (2001). *Econometric Analysis of Panel Data*, 3rd edition. John Wiley and Sons Ltd.  
 Baltagi BH (2013). *Econometric Analysis of Panel Data*, 5th edition. John Wiley and Sons Ltd.  
 Baltagi BH, Pinnoi N (1995). "Public capital stock and state productivity growth: further evidence from an error components model." *Empirical Economics*, **20**, 351-359.  
 Munnell A (1990). "Why Has Productivity Growth Declined? Productivity and Public Investment." *New England Economic Review*, 3-22.

---

|            |                                       |
|------------|---------------------------------------|
| unfold_idx | <i>Fold and Unfold a dfidx object</i> |
|------------|---------------------------------------|

---

## Description

fold\_idx takes a dfidx, includes the indexes as stand alone columns, remove the idx column and return a data.frame, with an ids attribute that contains the informations about the indexes. fold\_idx performs the opposite operation

## Usage

```
unfold_idx(x)

fold_idx(x, pkg = NULL)
```

## Arguments

|     |                                               |
|-----|-----------------------------------------------|
| x   | a dfidx object                                |
| pkg | if not NULL, this argument is passed to dfidx |

## Value

a data.frame for the unfold\_dfidx function, a dfidx object for the fold\_dfidx function

## Author(s)

Yves Croissant

## Examples

```
mn <- dfidx(munnell, idx = c(region = "state", "year"), position = 3, name = "index")
mn2 <- unfold_idx(mn)
attr(mn, "ids")
mn3 <- fold_idx(mn2)
identical(mn, mn3)
```

# Index

- \* **datasets**
  - munnell, [9](#)
- \* **dataset**
  - munnell, [9](#)
  - `[.dfidx (methods.dfidx),` [6](#)
  - `[[.dfidx (methods.dfidx),` [6](#)
  - `[[<-.dfidx (methods.dfidx),` [6](#)
  - `$.dfidx (methods.dfidx),` [6](#)
  - `$<-.dfidx (methods.dfidx),` [6](#)
  - `as.data.frame.dfidx (methods.dfidx),` [6](#)
  - `dfidx,` [2](#)
  - `fold_idx (unfold_idx),` [10](#)
  - `format_idx (idx),` [4](#)
  - `head.dfidx (methods.dfidx),` [6](#)
  - `idx,` [4](#)
  - `idx_name,` [5](#)
  - `mean.dfidx (methods.dfidx),` [6](#)
  - `methods.dfidx,` [6](#)
  - `model.frame.dfidx,` [8](#)
  - `model.matrix.dfidx (model.frame.dfidx),`  
[8](#)
  - munnell, [9](#)
  - `munnell_wide (munnell),` [9](#)
  - `print.dfidx (methods.dfidx),` [6](#)
  - `print.dfidx_matrix (model.frame.dfidx),`  
[8](#)
  - `print.idx (methods.dfidx),` [6](#)
  - `print.xseries (methods.dfidx),` [6](#)
  - `unfold_idx,` [10](#)