

Package: dfix (via r-universe)

October 21, 2024

Version 0.1-0

Date 2024-08-22

Title Indexed Data Frames

Depends R (>= 3.5.0)

Imports dplyr, Formula, vctrs, pillar, glue, Rdpack, tidyselect

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=dfix>

VignetteBuilder quarto

RoxygenNote 7.3.1

Roxygen list(markdown = TRUE)

Encoding UTF-8

LazyData true

RdMacros Rdpack

Repository <https://ycroissant.r-universe.dev>

RemoteUrl <https://github.com/ycroissant/dfix>

RemoteRef HEAD

RemoteSha a21e7dbef24d82cc9c1bfacf20d9ff8b08d096ad

Contents

dfix	2
dplyr	4
idx	5
idx_name	6
methods.dfix	7
model.frame.dfix	9
unfold_idx	10

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 therefore 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

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

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

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

```

dplyr

Methods for dplyr verbs

Description

methods of dplyr verbs for dfix objects. Default functions don't work because most of these functions returns either a tibble or a data.frame but not a dfix

Usage

```

## S3 method for class 'dfix'
arrange(.data, ...)

## S3 method for class 'dfix'
filter(.data, ...)

## S3 method for class 'dfix'
slice(.data, ...)

## S3 method for class 'dfix'
mutate(.data, ...)

## S3 method for class 'dfix'
transmute(.data, ...)

## S3 method for class 'dfix'
select(.data, ...)

```

Arguments

<code>.data</code>	a dfix object,
<code>...</code>	further arguments

Details

These methods always return the data frame column that contains the indexes and return a dfix object.

Value

an object of class "dfidx"

Author(s)

Yves Croissant

Examples

```
mn <- dfidx(munnell)
select(mn, - gsp, - water)
mutate(mn, lgsp = log(gsp), lgsp2 = lgsp ^ 2)
transmute(mn, lgsp = log(gsp), lgsp2 = lgsp ^ 2)
arrange(mn, desc(unemp), labor)
filter(mn, unemp > 10)
pull(mn, gsp)
slice(mn, c(1:2, 5:7))
```

idx

Index for dfidx

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

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

Details

idx is defined as a generic with a dfix 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 <- dfix(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 'dfix'
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 = 10L)

## 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 = 10L)

## 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 dfix
data	a formula
..., lhs, rhs, dot	see the Formula method
alt.subset	a subset of levels for the second index
reflevel	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 dfix object
x	a model matrix
n	the number of lines to print

Value

a dfix object for the model.frame method and a matrix for the model.matrix method.

Author(s)

Yves Croissant

Examples

```
mn <- dfix(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)
```

unfold_idx

Fold and Unfold a dfix object

Description

fold_idx takes a dfix, 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

`[.dfidx (methods.dfidx), 7`
`[[.dfidx (methods.dfidx), 7`
`[[<-.dfidx (methods.dfidx), 7`
`$.dfidx (methods.dfidx), 7`
`$<-.dfidx (methods.dfidx), 7`

`arrange.dfidx (dplyr), 4`
`as.data.frame.dfidx (methods.dfidx), 7`

`dfidx, 2`
`dplyr, 4`

`filter.dfidx (dplyr), 4`
`fold_idx (unfold_idx), 10`
`format.idx (idx), 5`

`head.dfidx (methods.dfidx), 7`

`idx, 5`
`idx_name, 6`

`mean.dfidx (methods.dfidx), 7`
`methods.dfidx, 7`
`model.frame.dfidx, 9`
`model.matrix.dfidx (model.frame.dfidx),
9`
`mutate.dfidx (dplyr), 4`

`print.dfidx (methods.dfidx), 7`
`print.dfidx_matrix (model.frame.dfidx),
9`
`print.idx (methods.dfidx), 7`
`print.xseries (methods.dfidx), 7`

`select.dfidx (dplyr), 4`
`slice.dfidx (dplyr), 4`

`transmute.dfidx (dplyr), 4`

`unfold_idx, 10`