Package ‘timbr’

April 29, 2023

Type Package
Title Forest/Tree Data Frames
Version 0.2.2
Description Provides data frames for forest or tree data structures. You can create forest data structures from data frames and process them based on their hierarchies.
License MIT + file LICENSE
Encoding UTF-8
RoxygenNote 7.2.3
Imports cli, dplyr, lifecycle, memoise, pillar, purrr, rlang, tibble, tidygraph, vctrs (>= 0.5.2)
Suggests covr, testthat (>= 3.0.0)
Config/testthat/edition 3
URL https://github.com/UchidaMizuki/timbr,

https://uchidamizuki.github.io/timbr/

BugReports https://github.com/UchidaMizuki/timbr/issues
NeedsCompilation no
Author Mizuki Uchida [aut, cre]
Maintainer Mizuki Uchida <uchidamizuki@vivaldi.net>
Repository CRAN
Date/Publication 2023-04-29 10:40:02 UTC

R topics documented:

   timbr-package ........................................ 2
   as_forest ........................................... 2
   children .......................................... 3
   climb .............................................. 3
   dplyr .............................................. 4
as_forest

Description

timbr: Forest Data Frames

Usage

as_forest(x, ...)

## S3 method for class 'rowwise_df'
as_forest(x, ...)

## S3 method for class 'grouped_df'
as_forest(x, ...)

Arguments

x An object.

... Unused, for extensibility.

Value

A forest.
children

---

**Description**

Convert a forest into a forest consisting of its child nodes.

**Usage**

```r
children(data, name = NULL)
```

**Arguments**

- `data`: A forest.
- `name`: ‘NULL’ (default) or a scalar character specifying the node name of child nodes.

**Value**

A forest.

---

climb

---

**Description**

Climb a forest from parents to children with one or more node names.

**Usage**

```r
climb(.data, ..., .deep = TRUE)
```

**Arguments**

- `.data`: A forest.
- `...`: A list of node names to climb the forest.
- `.deep`: Whether to search deeply for node names or not?

**Value**

A forest.
Description

dplyr methods for forest objects.

Usage

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

## S3 method for class 'forest'
summarise(.data, ..., .node = NULL)

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

## S3 method for class 'forest'
relocate(.data, ...)

## S3 method for class 'forest'
rows_update(x, y, by = NULL, ...)

## S3 method for class 'forest'
rows_patch(x, y, by = NULL, ...)

## S3 method for class 'forest'
rowwise(data, ...)

## S3 method for class 'forest'
ungroup(x, ...)

Arguments

.data A forest.
...

.node ‘NULL’ (default) or a vector to create new nodes.

by An unnamed character vector giving the key columns.

data A forest.

Value

A forest.
forest_by

Constructs a forest by one or more variables

Description

‘forest_by()’ constructs a forest by one or more variables.

Usage

forest_by(.data, ...)

Arguments

.data A data frame.
... Variables.

Value

A forest.

is_forest

Test if an object is a forest

Description

Test if an object is a forest

Usage

is_forest(x)

Arguments

x An object.

Value

‘TRUE’ if an object inherits from ‘forest’ class.
leaves  
*Leaf nodes of a forest*

### Description
Leaf nodes of a forest

### Usage
`leaves(data)`

### Arguments
- `data`: A forest.

### Value
A forest.

---

map_forest  
*Apply a function hierarchically to a forest*

### Description
Apply a function hierarchically to a forest in the climbing or descending direction.

### Usage
`map_forest(.x, .f, ..., .climb = FALSE)`

### Arguments
- `.x`: A forest
- `.f`: A function, formula, or vector (not necessarily atomic).
- `...`: Additional arguments passed on to the mapped function.
- `.climb`: Climbing or descending?

### Value
A forest.
### Attributes of root nodes

**Description**

Attributes of root nodes

**Usage**

- `node_name()`
- `node_value()`
- `node_parent()`

**Value**

A vector of names, values, or parents of root nodes.

---

### `traverse`

**Description**

Apply a function hierarchically to a forest in the climbing or descending direction.

**Usage**

`traverse(.x, .f, ..., .climb = FALSE)`

**Arguments**

- `.x` A forest
- `.f` A function, formula, or vector (not necessarily atomic).
- `...` Additional arguments passed on to the mapped function.
- `.climb` Climbing or descending?

**Value**

A forest.
Index

as_forest, 2
children, 3
climb, 3
dplyr, 4
forest_by, 5
is_forest, 5
leaves, 6
map_forest, 6
mutate.forest (dplyr), 4
node, 7
node_name (node), 7
node_parent (node), 7
node_value (node), 7
relocate.forest (dplyr), 4
rows_patch.forest (dplyr), 4
rows_update.forest (dplyr), 4
rowwise.forest (dplyr), 4
select.forest (dplyr), 4
summarise.forest (dplyr), 4
timbr-package, 2
traverse, 7
ungroup.forest (dplyr), 4