Package: BIOdry (via r-universe)

October 30, 2026

Type Package

Title Multilevel Modeling of Dendroclimatical Fluctuations

Version 0.9.6

Date 2026-10-29

Maintainer Wilson Lara-Henao <wilarhen@gmail.com>

Description Processes multilevel ecological data series (MEDS) arranged in temporal or spatial hierarchies defined by sampling designs. The package recursively applies one-level stage functions to long-format tree-ring and climate data, then optionally fits a terminal model to the processed series. Terminal fits currently include mixed-effects detrending through 'nlme' and Friedman supersmoother detrending through supsmu() of 'stats'. The package also supports post-detrending comparison and inference tools for multilevel dendroclimatic analyses. The method is described in Lara et al. (2018) <doi:10.1371/journal.pone.0196923>.

License GPL-3

Depends R (>= 4.1.0), nlme, ecodist

Imports lattice

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

LazyData TRUE

NeedsCompilation no

Author Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Repository https://cran.r-universe.dev

Date/Publication 2026-10-30 19:31:02 UTC

RemoteUrl https://github.com/cran/BIOdry

RemoteRef HEAD

RemoteSha 745830fb38ea1497ed0c4419b70550ba84807300

Contents

BIOdry-package . . . 3
amod . . . 5
anova.frametoLme . . . 6
anova.modelFrame . . . 8
arguSelect . . . 9
cClass . . . 11
defineDataRoles . . . 12
frametoFriedman . . . 14
frametoLme . . . 15
FriedmanForm . . . 18
levexp . . . 19
lmeForm . . . 20
modelFrame . . . 22
moveYr . . . 24
muleMan . . . 25
mUnits . . . 27
normalizeGroupLevels . . . 28
Pchron . . . 29
plot.frametoFriedman . . . 30
plot.frametoLme . . . 32
plot.modelFrame . . . 33
plot.muleMan . . . 34
plot.wlai . . . 35
Pradii03 . . . 37
Prec . . . 38
Prings05 . . . 38
print.frametoFriedman . . . 39
print.frametoLme . . . 40
print.modelFrame . . . 41
print.summary.frametoFriedman . . . 42
print.summary.frametoLme . . . 43
print.summary.modelFrame . . . 44
PTclim05 . . . 45
ringApply . . . 46
rtimes . . . 49
scacum . . . 50
shiftFrame . . . 51
slitFrame . . . 53
summary.frametoFriedman . . . 54
summary.frametoLme . . . 55
summary.modelFrame . . . 56
tdForm . . . 57
Temp . . . 58
update.frametoLme . . . 59
wlai . . . 60

Index **63**

BIOdry-package *Multilevel Modeling of Dendroclimatical Fluctuations*

Description

Multilevel ecological data series (MEDS) are sequences of observations ordered according to temporal/spatial hierarchies that are defined by sample designs, with sample variability confined to ecological factors. Dendroclimatic MEDS of tree rings and climate are modeled into normalized fluctuations of tree growth and aridity. Modeled fluctuations (model frames) are compared with Mantel correlograms on multiple levels defined by sample design. Package implementation can be understood by running examples in modelFrame(), and muleMan() functions.

Details

The DESCRIPTION file:

Package:	BIOdry
Type:	Package
Title:	Multilevel Modeling of Dendroclimatical Fluctuations
Version:	0.9.6
Date:	2026-10-29
Authors@R:	c(person(given = "Wilson", family = "Lara-Henao", role = c("aut", "cre"), email = "wilarhen@gmail
Maintainer:	Wilson Lara-Henao <wilarhen@gmail.com>
Description:	Processes multilevel ecological data series (MEDS) arranged in temporal or spatial hierarchies defin
License:	GPL-3
Depends:	R (>= 4.1.0), nlme, ecodist
Imports:	lattice
Suggests:	testthat (>= 3.0.0)
Config/testthat/edition:	3
LazyData:	TRUE
NeedsCompilation:	no
Author:	Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [au

Index of help topics:

BIOdry-package	Multilevel Modeling of Dendroclimatical Fluctuations
FriedmanForm	Friedman terminal specification
PTclim05	Multilevel data set of monthly cumulative precipitations and monthly average temperatures
Pchron	Tree-ring width (TRW) chronology
Pradii03	Vector of radii measured on P.pinaster
Prec	Cumulative precipitations
Prings05	Multilevel data set of P. pinaster
Temp	Average temperatures
amod	amod

anova.frametoLme	Compare frametoLme objects
anova.modelFrame	Compare modelFrame objects
arguSelect	Argument selection
cClass	Column-class extraction
defineDataRoles	defineDataRoles
frametoFriedman	Friedman supersmoother detrending
frametoLme	LME modeling
levexp	levexp
lmeForm	LME formula
mUnits	Metric system
modelFrame	Recursive dispatch of multilevel series functions
moveYr	Seasonal years
muleMan	Multilevel dendroclimatic correlograms
normalizeGroupLevels	Order grouping levels
plot.frametoFriedman	Plot Friedman fluctuation series over time
plot.frametoLme	Plot frametoLme objects
plot.modelFrame	Plot modelFrame objects
plot.muleMan	Plot muleMan objects
plot.wlai	Plot Walter-Lieth climate diagrams
print.frametoFriedman	Print frametoFriedman objects
print.frametoLme	Print frametoLme objects
print.modelFrame	Print modelFrame objects
print.summary.frametoFriedman	Print compact frametoFriedman summaries
print.summary.frametoLme	Print compact frametoLme summaries
print.summary.modelFrame	Print compact modelFrame summaries
ringApply	Multilevel apply
rtimes	Time-units synchronization
scacum	scacum
shiftFrame	shiftFrame
slitFrame	slitFrame
summary.frametoFriedman	Summarize frametoFriedman objects
summary.frametoLme	Summarize frametoLme objects
summary.modelFrame	Summarize modelFrame objects
tdForm	ltd formulas
update.frametoLme	Update frametoLme objects
wlai	Walter-Lieth aridity index

Maintainer: Wilson Lara-Henao <wilarhen@gmail.com>

Author(s)

Wilson Lara-Henao <wilarhen@gmail.com>, Felipe Bravo <fbravo@pvs.uva.es>

amod *amod*

Description

Allometric scaling. Allometric models and parameters are used to scale organic growth.

Usage

```
amod(x, mp = c(1, 1),
    fun = y ~ a * (x^b),
    lv = NULL, validate = c("none",
        "warn", "stop"))
```

Arguments

x	numeric vector or «data.frame.
mp	numeric. Allometric parameters. Several groups may be supplied recursively.
fun	formula.
lv	NULL, character, «or numeric. Grouping level(s) to split multilevel data frames before scaling.
validate	Validation mode.

Details

. This patched version keeps the original block-level helper behavior, but it also works on multilevel data frames. When a data frame is supplied with lv, the function first resolves the grouping hierarchy from lv, then uses slitFrame() to materialize those groups, orders rows within each block on the available time variables that remain after splitting, and finally applies the allometric model to each ordered block. Character and numeric lv values are both supported. Numeric lv values are interpreted as positions in the detected nesting structure, whereas character values are matched to grouping names directly. Cumulative organic growth is taken from csx when available. If csx is absent but x or trw is present, cumulative growth is derived with cumsum() after time ordering only for the internal scaling step, so the function can operate on simpler long-format tree-ring tables. When the allometric model is applied, the returned object always contains both the scaled increments (x) and the scaled cumulative growth state (csx). This keeps the transformed increment and state variables paired explicitly after scaling. The first transformed increment is the change from the size before the first ring (csx minus the first ring width, or zero when no ring widths are available) to the first transformed cumulative value, so identity settings such as mp = c(1,1) preserve the original increments rather than introducing an initial NA. Direct use on climatic tables is only appropriate when a meaningful cumulative source is available or can be derived before scaling. The returned data frame preserves the original row order after block-wise processing and keeps scaled outcomes before time and grouping columns.

Value

data.frame of the scaled variable (x) and cumulative scaled growth (csx). For multilevel data frames, these are computed within each ordered split block.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## Simulating TRW records:
set.seed(1)
trw <- ts(abs(rnorm(12,1,1)),start = 1950)
## Cumulative TRW:
cri <- cumsum(trw)
## tree diameters
td <- amod(cri,mp = c(2,1))
## plot of the tree diameters and the
## relative increments:
plot(ts(td))
```

---

anova.frametoLme *Compare frametoLme objects*

Description

Analysis-of-variance tables of the fixed effects of one frametoLme fit, or likelihood comparison of several, computed by anova.lme on the underlying lme fits.

Usage

```
## S3 method for class 'frametoLme'
anova(object,
    ..., test = TRUE,
    type = c("sequential",
        "marginal"),
    adjustSigma = TRUE,
    Terms, L, verbose = FALSE)
```

Arguments

object	A frametoLme fit, or an lme fit.
...	Further frametoLme or lme objects to compare.
test	logical. With several models, compute likelihood-ratio tests. Passed to anova.lme.
type	character. With one model, sequential or marginal F tests. See anova.lme.

adjustSigma	logical. With one model fitted by ML, adjust the residual standard error. See anova.lme.
Terms	With one model, the fixed-effects terms tested jointly, by name or position. See anova.lme.
L	With one model, a contrast matrix of the fixed effects. See anova.lme.
verbose	logical. Print the model calls. Passed to anova.lme.

Details

With one model, the call is forwarded to anova.lme on the stored lme fit, so sequential or marginal F tests, tests of chosen terms and contrast matrices work as in **nlme**. With several models, their lme fits are compared by AIC, BIC and log-likelihood, with likelihood-ratio tests when the models differ in their number of parameters; rows are named after the objects given. frametoLme and lme objects can be mixed. As in **nlme**, models that differ in their fixed effects are compared only when fitted by maximum likelihood (method = 'ML') to the same data; update.frametoLme refits a model with other arguments.

Value

The anova.lme table of anova.lme.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

References

Pinheiro J. C., D. M. Bates. 2000. Mixed-effects models in S and S-PLUS. Springer, New York.

Examples

```
## tree-ring widths (Prings05) and radii measured in 2003 (Pradii03);
## Zeide model with ln(time) and a tree-level random part, as in the
## frametoLme example
data(Prings05, envir = environment())
data(Pradii03, envir = environment())
fit <- Prings05 |>
    ringApply(lv = 2, fn = 'rtimes') |>
    ringApply(lv = 'sample', sc.c = Pradii03, rf.t = 2003,
              fn = 'scacum') |>
    frametoLme(form = 'log(x) ~ log(csx) + log(time) | tree',
               random = pdDiag(~ log(csx)))
## sequential and marginal F tests of the fixed effects
anova(fit)
anova(fit, type = 'marginal')
## F test of one term
anova(fit, Terms = 'log(time)')
## the two Zeide time forms, f(t) = ln t and f(t) = t, refitted by
## maximum likelihood and compared by AIC and BIC
```

```
fit_lnt <- update(fit, method = 'ML')
fit_t <- update(fit_lnt, form = 'log(x) ~ log(csx) + time | tree')
anova(fit_t, fit_lnt)
```

`anova.modelFrame` *Compare modelFrame objects*

Description

Models in `modelFrame` objects are compared with an ANOVA method when the terminal fit is based on `lme`.

Usage

```
## S3 method for class 'modelFrame'
anova(object,
    ..., test, type,
    adjustSigma, Terms,
    L, verbose)
```

Arguments

`object`	An `object` inheriting from class `modelFrame`.
`...`	Other optional fitted model objects inheriting from classes `"modelFrame"`, `"lme"`, `"lm"`, among others accepted by `anova.lme`.
`test`	test
`type`	type
`adjustSigma`	adjustSigma
`Terms`	Terms
`L`	L
`verbose`	verbose

Value

A data frame inheriting from class `"anova.lme"` when the underlying terminal fit is mixed-effects.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## ring-width chronologies of eight cores and the allometric
## parameters of the modelFrame example
data(Pchron, envir = environment())
biom_param <- c(2.87, 0.85, 0.05, 2.5)
## the biomass model with the two Zeide time forms, f(t) = t and
## f(t) = ln t, fitted by maximum likelihood with a tree intercept and a
## tree-level log(csx) slope
m_t <- Pchron |>
    modelFrame(to = 'cm', MoreArgs = list(mp = c(2, 1, biom_param)),
               form = 'log(x) ~ log(csx) + time | tree',
               random = pdDiag(~ log(csx)), method = 'ML')
m_lnt <- Pchron |>
    modelFrame(to = 'cm', MoreArgs = list(mp = c(2, 1, biom_param)),
               form = 'log(x) ~ log(csx) + log(time) | tree',
               random = pdDiag(~ log(csx)), method = 'ML')
## sequential F tests of the fixed effects of one model
anova(m_t)
## both forms have the same number of parameters: compare AIC and BIC
anova(m_t, m_lnt)
```

`arguSelect` *Argument selection*

Description

Arguments of specific functions are selected from arbitrary numbers and varieties of arguments.

Usage

```
arguSelect(rd = NULL,
    fun = c("mapply",
        "ringApply"),
    ..., validate = c("none",
        "warn", "stop"))
```

Arguments

`rd`	`NULL` or `data.frame`. Multilevel ecological data series. If `NULL` then this argument is ignored.
`fun`	`character` or `NULL`. Vector of function names.
`...`	Further arguments not necessarily contained in the processed function(s).
`validate`	validate

Details

Closures with ellipsis terms use this function to extract and pass arguments to other functions. Arguments in `MoreArgs` lists are also extracted and stored again as `MoreArgs` lists.

Value

`list` of selected arguments.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
##Multilevel ecological data series of tree-ring widths:
data(Prings05,envir = environment())
## Radial increments measured on 2003:
data(Pradii03,envir = environment())

## Selection of arguments in some functions:
ar1 <- arguSelect(fun = c('amod'),
                  only.dup = TRUE,
                  mp = c(0.5,1),
                  rf.t = 2003)
str(ar1)

ar2 <- arguSelect(fn = 'amod',
                  only.dup = TRUE,
                  mp = c(0.5,1),
                  rf.t = 2003)
str(ar2)
ar3 <- arguSelect(rd = Prings05,
                  fn = 'amod',
                  only.dup = TRUE,
                  mp = c(0.5,1),
                  rf.t = 2003)
str(ar3)

ar4 <- arguSelect(rd = Prings05,
                  fun = 'scacum',
                  sc.c = Pradii03,
                  MoreArgs = list(only.dup = TRUE,
                                  mp = c(0.5,1),
                                  rf.t = 2003))
str(ar4)

ar5 <- arguSelect(rd = Prings05,
                  fun = 'scacum',
                  ref = Pradii03,
                  rf.t = rep(2003:2011),
                  MoreArgs = list(only.dup = TRUE,
                                  mp = c(0.5,1)))
str(ar5)
```

cClass *Column-class extraction*

Description

Column names of multilevel data sets are extracted according to three classes, which BIOdry uses as column roles: `numeric` (numeric columns that are not whole numbers; responses), `integer` (whole-number numeric columns, whatever their storage mode; time candidates), and `factor` (factor and character columns; grouping candidates, except the month column).

Usage

```
cClass(rd, cl = "all",
    validate = c("none",
        "warn", "stop"))
```

Arguments

`rd`	`data.frame`. Multilevel data series.
`cl`	`character` or `NULL`. Character vector of classes to be considered. These can be 'numeric', 'integer', or 'factor'. If `'all'` then all column names of `rd` are extracted.
`validate`	Validation mode.

Value

`character` names.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
##Multilevel data frame of tree-ring widths:
data(Prings05,envir = environment())
## Names of variables in Prings05 data containing numeric classes:
cClass(Prings05, 'numeric') # 'x'
## Names of variables containing time units:
cClass(Prings05, 'integer') # 'year'
## Names of variables containing factors:
cClass(Prings05, 'factor') # 'sample', 'tree', 'plot'
```

defineDataRoles *defineDataRoles*

Description

Constructor for BIOdry role metadata Attach variable-role metadata to multilevel ecological data

Usage

```
defineDataRoles(rd, inputs = NULL,
    response = NULL,
    time = NULL, groups = NULL,
    name_map = NULL,
    validate = c("warn",
        "stop", "none"))
```

Arguments

rd	data.frame. Multilevel ecological or climatic data.
inputs	Optional character vector of raw input or measurement variables. For climate data this can contain more than one variable, for example c("pr", "tm"); for wlai precipitation comes first (columns named pr and tm are used by name). If omitted, numeric non-group, non-time columns are inferred.
response	Optional character scalar giving the current active response variable. If omitted and exactly one input variable is inferred, that variable is used as the response; otherwise the response remains unset.
time	Optional character vector naming one or more ordered time variables. Examples include "year" or c("month", "year"). The constructor stores all supplied names in time_vars and the first one as the primary time variable in time.
groups	Optional grouping hierarchy, either as a character vector or as a nested string with / separators such as "plot/tree/sample". If omitted, the factor and character columns other than the month column are inferred, outer level first.
name_map	Optional named character vector mapping older variable names to current names, for example c(year = "hydro_year_id"). This mapping is stored verbatim and can be updated by later workflow steps.
validate	Validation mode. Use "warn" to keep partial metadata with a diagnostic, "stop" to require a fully resolved declaration, or "none" for quiet inference.

Details

This constructor attaches a small BIOdry_meta attribute to a data.frame so that later workflow steps can use declared roles. Functions such as lmeForm, tdForm, frametoLme, and FriedmanForm decide roles from column types (see cClass); a time, groups or response declared here takes precedence over that rule. Roles that are only inferred, and the metadata that ringApply records automatically, do not.

The constructor respects explicit user definitions first. If the user supplies `time`, `groups`, `inputs`, or `response`, these values are stored as given after validation. If they are not supplied, the constructor infers them from column types: the `time` variable is the first whole-number numeric column, the `groups` are the factor and character columns other than the month column, outer level first, and the `inputs` are the remaining numeric columns. A `time` variable is not also placed in the grouping hierarchy unless the user includes it in `groups`.

The `time` argument may be given as a single variable or as a character vector describing ordered `time` components such as `c("month", "year")`. In that case, all supplied names are stored in `time_vars`, while a conservative primary `time` variable, the first supplied element, is also stored in `time` for downstream functions that expect a single covariate name.

The `groups` argument may be given as a character vector such as `c("plot", "tree", "sample")` or as a nested string such as `"plot/tree/sample"`. Whole-number columns, such as `year`, are grouping levels only when declared here; to use `year` as a random grouping level, see `frametoLme`.

The constructor is intentionally light. It does not copy the data, change user-visible columns, or fit any models. It only stores compact metadata describing: raw input measurement variables, the current active `response` when available, the primary `time` variable, all declared `time` components, the grouping hierarchy, whether `time` was also declared as part of that hierarchy, and an optional `name_map` linking older names to derived names such as `year -> hydro_year_id`. This allows later functions to preserve context across successive transformations.

When automatic inference is not possible, the function fails gracefully according to `validate`. Use `validate = "warn"` to see a diagnostic while still returning the data with partial metadata, or `validate = "stop"` to require a fully resolved declaration.

Value

`data.frame` with a `BIOdry_meta` attribute.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## tree-ring widths: declare the time and the grouping hierarchy
data(Prings05, envir = environment())
dr1 <- defineDataRoles(Prings05, time = "year",
                       groups = "plot/tree/sample")
str(attr(dr1, "BIOdry_meta"))
## declared roles are used by the formula helpers
lmeForm(dr1)

## monthly climate: precipitation first, then temperature, as wlai()
## reads them; year and month as time components
data(PTclim05, envir = environment())
cl1 <- defineDataRoles(PTclim05, inputs = c("pr", "tm"),
                       time = c("year", "month"), groups = "plot")
str(attr(cl1, "BIOdry_meta"))
```

`frametoFriedman` *Friedman supersmoother detrending*

Description

Friedman smoothers are fitted to multilevel ecological series.

Usage

```
frametoFriedman(rd, form = "FriedmanForm",
    response = NULL,
    time = NULL, groups = NULL,
    res.data = TRUE,
    difference = NULL,
    span = NULL, bass = NULL,
    periodic = NULL,
    min.obs = NULL, validate = c("none",
        "warn", "stop"),
    ...)
```

Arguments

`rd`	`data.frame`. Multilevel ecological data series.
`form`	`character`, `list`, or `NULL`. Terminal smoother specification.
`response`	`NULL` or `character`. Response series.
`time`	`NULL` or `character`. Time variable.
`groups`	`NULL` or `character`. Grouping variables.
`res.data`	`logical`. If `TRUE`, attach detrended series in component `fluc`.
`difference`	`NULL` or `logical`. Use subtraction instead of ratio detrending.
`span`	`NULL`, `character`, or `numeric`. Span.
`bass`	`NULL` or `numeric`. Bass argument.
`periodic`	`NULL` or `logical`. Periodic option.
`min.obs`	`NULL` or `integer`. Minimum finite observations per series.
`validate`	`character`. Validation mode.
`...`	Unused extra arguments kept for compatibility.

Details

This terminal fitter applies `supsmu` series by series on a multilevel long-format data frame. It resolves one `response` variable, one primary `time` variable, and a grouping hierarchy defining the independent series.

This patched version is constructor-aware. When a BIOdry_meta attribute is available, the resolved time variable is excluded from the Friedman grouping hierarchy unless time was explicitly declared as part of groups. This prevents accidental smoothing of one-point series such as zone / hydro_year_id when the intended fit is one series per zone over hydro_year_id.

The returned object is still a Friedman detrending object, not a mixed model. If res.data = TRUE, component $fluc contains detrended series and fitted values for plotting and later second-stage analyses.

Value

list of class "frametoFriedman" with at least components model and call, and optionally fluc.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

References

Friedman, J. H. 1984. A variable span smoother. Laboratory for Computational Statistics, Stanford University Technical Report 5.

Examples

```
## tree-ring widths (Prings05) and radii measured in 2003 (Pradii03):
## relative time within trees, then cumulative growth scaled to the radii
data(Prings05, envir = environment())
data(Pradii03, envir = environment())
dfm2 <- Prings05 |>
    ringApply(lv = 2, fn = 'rtimes') |>
    ringApply(lv = 'sample', sc.c = Pradii03, rf.t = 2003,
              fn = 'scacum')
## Friedman supersmoother on each core; ratio index x / smooth
ff <- frametoFriedman(dfm2)
ff
head(ff$fluc)
## difference index x - smooth
frametoFriedman(dfm2, difference = TRUE)$fluc |> head()
```

---

frametoLme *LME modeling*

Description

LME models are fitted to detrend multilevel ecological data series.

Usage

```
frametoLme(rd, form = "lmeForm",
    res.data = TRUE,
    validate = c("none",
        "warn", "stop"),
    ...)
```

Arguments

rd	data.frame. Multilevel ecological data series.
form	character. Any of two lme formulas: "lmeForm" and "tdForm". A formula string containing ~ is used directly. If NULL, the input data are returned unchanged.
res.data	logical. Save residuals as a multilevel ecological data series. If TRUE then a data frame of name "fluc" is added to output. Its residuals are matched to the model-data rows by row label, so each residual keeps its own time and grouping values.
validate	character. Optional validation mode.
...	Further arguments passed to lme or to the formula helper named in form.

Details

This function implements lme to fit linear mixed-effects models on multilevel ecological data series. When form is a character without a tilde it is resolved through the named formula helper, typically lmeForm or tdForm; when it already contains a tilde it is used directly as a formula. When form = NULL, the function now returns the processed input data unchanged, which makes direct ringApply() pipelines safe. Before model fitting, the function resolves the final formula and keeps only the variables used by that formula and the grouping columns. Rows with a missing value in any of them are dropped from the fit without a message (use validate = "warn" to be told how many). For example, after ringApply(Prings05, lv = 2, fn = "rtimes") the 16 rows where rtimes gives no time are not in a fit that uses time, so the model has 992 of the 1008 rows. Missing values in unused columns, such as an alternative time index, do not drop rows. The returned object receives class "frametoLme", which enables dedicated print() and summary() methods.

Unless a random part is passed through ..., it is pdDiag of the whole right-hand side of the fixed formula: every fixed term, including covariates written into a formula string, gets an uncorrelated random effect at each grouping level. When a level has very few groups, the default random part can give a singular fit (variance components near zero) with inflated standard errors of the fixed effects; a simpler random part can then be given through .... In the example, the bundled data (2 plots x 2 trees x 2 cores) support a tree intercept and a tree-level log(csx) slope only.

With form = "tdForm", the time term is linear by default or logarithmic with log.t = TRUE (passed through ...); both are Zeide forms with the same number of parameters, and the example compares them by AIC under method = "ML". Within cores, log(csx) and the time term are strongly correlated (r about 0.95 for time and 0.99 for ln(time) on the bundled data), so their separate standard errors depend on the random part.

Whole-number columns such as year are treated as time, not as groups (see cClass). To use year as a random grouping level of a data frame d, either declare it, as in defineDataRoles(d, time =

"time", groups = c("year", "plot", "tree","sample")), or add a factor copy, as in d$fyear <- factor(d$year), and give frametoLme() a formula string whose grouping part names it, such as "log(x) ~ log(csx) + time | year". A factor copy becomes a grouping column in the default formulas, so name the grouping levels explicitly or remove the copy with lev.rm.

Value

list of class "frametoLme" with at least components model and call, and optionally fluc: Pearson residuals at every grouping level, each matched by row label to its row of the model data and returned with that row's time and grouping columns. If a correlation or variance structure is passed to lme through ..., these remain Pearson residuals, not normalized residuals.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

References

Pinheiro J. C., D. M. Bates. 2000. Mixed-effects models in S and S-PLUS. Springer, New York.

Examples

```
## tree-ring widths (Prings05) and radii measured in 2003 (Pradii03):
## relative time within trees, then cumulative growth scaled to the radii
data(Prings05, envir = environment())
data(Pradii03, envir = environment())
dfm2 <- Prings05 |>
    ringApply(lv = 2, fn = 'rtimes') |>
    ringApply(lv = 'sample', sc.c = Pradii03, rf.t = 2003,
              fn = 'scacum')
## the bundled data are a small demonstration set (2 plots x 2 trees x
## 2 cores) that supports variation among trees only: a tree intercept and
## a tree-level log(csx) slope (tree labels are unique across plots here,
## otherwise nest them, | plot/tree)
## both time forms are Zeide forms, f(t) = t (log.t = FALSE, the default)
## and f(t) = ln t (log.t = TRUE); fit both with ML and let the fit choose
fit_t <- dfm2 |>
    frametoLme(form = 'log(x) ~ log(csx) + time | tree',
               random = pdDiag(~ log(csx)), method = 'ML')
fit_lnt <- dfm2 |>
    frametoLme(form = 'log(x) ~ log(csx) + log(time) | tree',
               random = pdDiag(~ log(csx)), method = 'ML')
anova(fit_t, fit_lnt)
## here the data prefer f(t) = ln t (lower AIC and BIC); refit it with REML
fit <- dfm2 |>
    frametoLme(form = 'log(x) ~ log(csx) + log(time) | tree',
               random = pdDiag(~ log(csx)))
head(fit$fluc)
print(summary(fit))
```

FriedmanForm *Friedman terminal specification*

Description

This function formulates Friedman smoother specifications for multilevel ecological data series.

Usage

```
FriedmanForm(rd, response = NULL,
    time = NULL, groups = NULL,
    difference = FALSE,
    span = "cv", bass = 0,
    periodic = FALSE,
    min.obs = 4L, validate = c("none",
        "warn", "stop"))
```

Arguments

rd	data.frame. Multilevel ecological data series.
response	NULL or character. Name of the response series to smooth.
time	NULL or character. Name of the time variable. A vector is allowed; its first element present in rd is used.
groups	NULL or character. Columns that define individual series. Nested strings such as "plot/tree/sample" are accepted.
difference	logical. Use subtraction instead of ratio detrending.
span	character or numeric. Span argument passed to stats::supsmu().
bass	numeric. Bass argument passed to stats::supsmu().
periodic	logical. Periodic smoother option.
min.obs	integer. Minimum finite observations per series.
validate	character. Validation mode.

Details

Unlike tdForm and lmeForm, this helper does not return a regression formula. It returns a compact terminal specification used by frametoFriedman and modelFrame to identify the response series, the time variable, the grouping columns, and the main smoothing settings.

The time variable is the first whole-number numeric column and the grouping columns are the factor and character columns other than the month column (see cClass). Roles declared with defineDataRoles take precedence; the metadata that ringApply records automatically does not. The time variable is excluded from the grouping columns unless the declared metadata says that time also belongs to groups, so each series is the response over time within groups, not within group-time combinations with fewer than min.obs observations.

Value

`list` inheriting from class `"FriedmanForm"`.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

References

Friedman, J. H. 1984. A variable span smoother. Laboratory for Computational Statistics, Stanford University Technical Report 5.

Examples

```
## tree-ring widths (Prings05) and radii measured in 2003 (Pradii03):
## relative time within trees, then cumulative growth scaled to the radii
data(Prings05, envir = environment())
data(Pradii03, envir = environment())
dfm2 <- Prings05 |>
    ringApply(lv = 2, fn = 'rtimes') |>
    ringApply(lv = 'sample', sc.c = Pradii03, rf.t = 2003,
              fn = 'scacum')
## the specification resolved from the data: response, time, series
## and smoothing settings
FriedmanForm(dfm2) |> str()
## a difference index (x - smooth) instead of a ratio index (x / smooth)
FriedmanForm(dfm2, difference = TRUE)$difference
```

`levexp` *levexp*

Description

Fast and compatibility-oriented vector releveling Expands or reduces a named numeric vector by matching its level names to split labels from multilevel ecological data.

Usage

```
levexp(x, levels, validate = c("none",
    "warn", "stop"))
```

Arguments

`x`	named numeric vector with level names to be matched.
`levels`	data.frame, split list, or character vector of split names.
`validate`	validation mode.

Value

numeric vector with expanded/reduced `levels`.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
data(Prings05, envir = environment())
data(Pradii03, envir = environment())

refs <- levexp(Pradii03, Prings05)
refs
```

`lmeForm` *LME formula*

Description

This function computes LME formulas from multilevel ecological data series (MEDS).

Usage

```
lmeForm(rd, prim.cov = FALSE,
    resp = NULL, covar = NULL,
    lev.rm = NULL, validate = c("none",
        "warn", "stop"))
```

Arguments

`rd`	`data.frame`. Multilevel ecological data series
`prim.cov`	`Logical`: should the LME formula only be printed in primary covariate form: '~ cov'? If FALSE then a complete form: 'resp ~ `covar` | group' is formulated.
`resp`	`NULL` or `character`. Column name of the response. If NULL then a response declared with `defineDataRoles` is used, and otherwise the first non-integer numeric column of the MEDS.
`covar`	`NULL` or `character`. Column name(s) of the covariate(s). If `NULL` then a time variable declared with `defineDataRoles` is used, and otherwise the first whole-number numeric column of the MEDS.
`lev.rm`	`NULL`, `character` or `numeric` vector of levels in the MEDS to be removed from the groups.
`validate`	validate

Details

Formulas of the form `resp ~ cov | group` (see `groupedData` function) are computed from MEDS. The formulas can be implemented by `modelFrame` function to detrend MEDS.

Column roles come from column types (see `cClass`): the response is the first non-integer numeric column, the covariate is the first whole-number numeric column, and the grouping levels are the factor and character columns, outer level first (the reverse of the column order). The month column (named `month`, or a factor whose levels are month names) is never a grouping level.

A response, time covariate or grouping levels declared with `defineDataRoles` take precedence. The metadata that `ringApply` records automatically does not override the type rule. A declared time variable is removed from the grouping levels unless the metadata sets `time_in_groups = TRUE`. To use a whole-number column such as `year` as a grouping level, see `frametoLme`.

Value

`formula` with any of the forms: `resp ~ cov | group` or `~ cov`.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

References

Pinheiro J. C., D. M. Bates. 2000. Mixed-effects models in S and S-PLUS. Springer, New York.

Examples

```
##Multilevel ecological data series of tree-ring widths:
data(Prings05,envir = environment())

## LME formula:
form1 <- lmeForm(Prings05,prim.cov = FALSE)
print(form1)
## removing the sample level from the formula
form2 <- lmeForm(Prings05,lev.rm = 'sample')
form2 <- lmeForm(Prings05,lev.rm = 1)

## groupedData object with the LME formula
gdata <- groupedData(lmeForm(Prings05,lev.rm = 1),
                     data = Prings05)
plot(gdata,groups = ~ sample)
```

modelFrame *Recursive dispatch of multilevel series functions*

Description

This function recursively dispatches user-defined stage functions on multilevel long-format series and can optionally apply a terminal fitter specified by 'form'.

Usage

```
modelFrame(rd, fn = list("rtimes",
    "scacum", "amod"),
    lv = list(2, 1, 1),
    form = "tdForm",
    validate = c("none",
        "warn", "stop"),
    ...)
```

Arguments

rd	data.frame or list. Dendroclimatic chronology or multilevel ecological data series.
fn	list. Names of recursive stage functions to dispatch through ringApply. Stage functions must return a data.frame. The preferred grammar keeps terminal fitting out of fn; however, a final frametoLme or frametoFriedman entry is still recognized for backward compatibility.
lv	list. Splitting levels for the functions that are evaluated through ringApply. Numeric values are resolved against the current hierarchy at each stage. Character values are interpreted as grouping columns in the current data. If a terminal fitter is supplied as the last entry of fn for backward compatibility, lv applies only to the preceding stage functions.
form	NULL, character, formula, or list. Terminal model specification. NULL returns only the processed long data. tdForm, lmeForm, a formula, or a formula-like character string trigger frametoLme. FriedmanForm or an object inheriting from class "FriedmanForm" trigger frametoFriedman.
validate	character. Validation mode.
...	Further arguments in mUnits, the recursive stage functions, the terminal fitter, or the terminal specification helper named in form.

Details

By default the workflow applies rtimes, scacum, and amod through ringApply, then uses tdForm as the terminal model specification for frametoLme. More generally, modelFrame behaves as a recursive dispatcher for multilevel ecological series. Functions listed in fn are evaluated stage by stage on the hierarchy defined by lv. The terminal path is now driven by form: NULL returns the processed long-format data.frame; tdForm, lmeForm, an explicit formula, or a formula-like character

string trigger frametoLme; and FriedmanForm triggers frametoFriedman. For backward compatibility, a terminal fitter placed as the last entry of fn is still recognized, but the preferred grammar is now to keep fn for recursive stage functions and form for the terminal model or smoother specification. The implementation is conservative with multilevel data: it prefers explicit time columns over row names, reshapes only when the input clearly looks wide, restricts unit conversion to record-like numeric columns, resolves numeric lv positions against the current hierarchy at each stage, and validates the stage contract. In the terminal frametoLme fit, rows with a missing value in the fitted variables or the grouping columns are dropped without a message; with the default tdForm these are the rows where rtimes gives no time (16 of the 1008 rows of Pchron). Arguments for lme, such as random = pdDiag(~ log(csx)) with a formula grouped by tree (a tree intercept and a tree-level log(csx) slope, as in the frametoLme example), are passed on to frametoLme. Arguments in modelFrame objects can be updated with update. Use validate = "warn" or validate = "stop" for stricter checks.

Value

Processed long-format data.frame if form = NULL; otherwise the object returned by the terminal fitter, with the originating modelFrame call attached and class \"modelFrame\" prepended.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

References

Lara W., F. Bravo, D. Maguire. 2013. Modeling patterns between drought and tree biomass growth from dendrochronological data: A multilevel approach. Agric. For. Meteorol., 178-179:140-151.

Examples

```
## ring-width chronologies (mm) of eight cores in two plots, one
## column per core
data(Pchron, envir = environment())
## allometric parameters: diameter inside bark (cm) to diameter at
## breast height (cm), then to total tree biomass (kg)
biom_param <- c(2.87, 0.85, 0.05, 2.5)

## rtimes, scacum and amod stage by stage, then a model on year with a
## random intercept per tree: these 2 plots x 2 trees x 2 cores support
## variation among trees only (tree labels are unique across plots here)
Pchron |>
    modelFrame(to = 'cm', MoreArgs = list(mp = c(2, 1, biom_param)),
               form = 'log(x) ~ log(csx) + year | tree',
               random = pdDiag(~ 1))

## the same stages with a Friedman smoother as the terminal step
Pchron |>
    modelFrame(form = 'FriedmanForm', to = 'cm',
               MoreArgs = list(mp = c(2, 1, biom_param)),
               response = 'x', time = 'year')
```

```
## only the processed long data
Pchron |>
    modelFrame(form = NULL, to = 'cm',
               MoreArgs = list(mp = c(2, 1, biom_param))) |>
    as.data.frame() |>
    head()
```

---

moveYr *Seasonal years*

Description

Monthly records in time-series replicates (usually of climate) are labeled for the years can begin in a month other than January.

Usage

```
moveYr(cd, ini.mnt = "Oct",
    label_year = c("end",
        "start"), keep_hydro_id = TRUE,
    hydro_id_name = "hydro_year_id",
    validate = c("none",
        "warn", "stop"))
```

Arguments

cd	`data.frame` of monthly records with the columns `year` and `month`, and any grouping or value columns; or a `numeric` vector of years named by month.
ini.mnt	`character` or `numeric`. First month of the hydrological year, as a month abbreviation, full month name or number from 1 to 12.
label_year	`character`. How hydrological years are labeled in the column `year`. `'end'` (default) labels each year by the calendar year in which it ends; `'start'` labels it by the calendar year in which it starts. The column `hydro_year_id` always holds the start year.
keep_hydro_id	`logical`. Add a column with the calendar year in which each hydrological year starts (data frames only).
hydro_id_name	`character`. Name of that column. Other functions of the package, such as `wlai`, find it only under the default name.
validate	`character`. What to do when month labels cannot be read, years are not whole numbers, several month columns are found, or the rows are not sorted by group, year and month: `'none'` (default) goes on silently (unreadable months become `NA`), `'warn'` warns and `'stop'` stops.

Details

character months as defined in month.abb or month.name. The columns are found by the package's column names: year, and month (month abbreviations, full month names, or integers 1:12); a factor whose levels are month names is also accepted as the month column.

Value

For a data frame, the input with the month column as a factor of month abbreviations, the column year relabeled by hydrological year (see label_year), and, by default, a column hydro_year_id with the start year of each hydrological year. The calendar year is not kept: with ini.mnt = 'Oct' and label_year = 'end', October to December 1955 are labeled 1956. It can be recovered as hydro_year_id for the months from ini.mnt to December and hydro_year_id + 1 for the months before ini.mnt.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## monthly precipitation and temperature of two plots (PTclim05)
data(PTclim05, envir = environment())
## hydrological years from October to September, labeled by the
## calendar year in which they end (label_year = 'end', the default)
cl <- PTclim05 |> moveYr()
head(cl)
## October to December 1955 belong to the hydrological year 1956
subset(cl, plot == 'P44005' & hydro_year_id == 1955 &
           month %in% c('Oct', 'Nov', 'Dec', 'Jan'))
## the calendar year is recovered from hydro_year_id (see Value)
cl$calendar_year <- cl$hydro_year_id +
    (match(as.character(cl$month), month.abb) < 10)
all(cl$calendar_year[order(cl$plot, cl$calendar_year, cl$month)] ==
    PTclim05$year[order(PTclim05$plot, PTclim05$year, PTclim05$month)])
## labels by the start year, and hydrological years from April
PTclim05 |> moveYr(label_year = 'start') |> head(3)
PTclim05 |> moveYr(ini.mnt = 'Apr') |> head(3)
```

---

muleMan *Multilevel dendroclimatic correlograms*

Description

Multivariate correlograms between TRW fluctuations and climatic fluctuations.

Usage

```
muleMan(rd, cd, rd.var = NULL,
    cd.var = NULL, ...)
```

Arguments

rd	dataframe or groupedData. TRW fluctuations such as that produced by modelFrame.
cd	dataframe or groupedData. Aridity-index fluctuations such as that produced by modelFrame.
rd.var	character or NULL. Column name of the TRW fluctuations to be compared. If NULL then the first column is processed.
cd.var	character or NULL. Column name of the aridity-index fluctuations to be compared. If NULL then the first column is used.
...	Further arguments in mgram

Details

Function mgram is implemented to compare the dendroclimatic fluctuations. Models being compared should have common higher-level factors (see example).

Value

data.frame object of multivariate correlations.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

References

Lara W., F. Bravo, D. Maguire. 2013. Modeling patterns between drought and tree biomass growth from dendrochronological data: A multilevel approach. Agric. For. Meteorol., 178-179:140-151.

Examples

```
## ring-width chronologies (mm) of eight cores and the allometric
## parameters of the modelFrame example
data(Pchron, envir = environment())
biom_param <- c(2.87, 0.85, 0.05, 2.5)
## tree-biomass series of each core, processed without a terminal
## model (on these 2 plots x 2 trees x 2 cores no mixed model with plot,
## tree and core levels is well posed), with the columns muleMan pairs
## on: the series, year and the plot, tree and core labels
trw <- Pchron |>
    modelFrame(form = NULL, to = 'cm',
               MoreArgs = list(mp = c(2, 1, biom_param)))
trw <- trw[, c('x', 'year', 'plot', 'tree', 'sample')]
## aridity-index fluctuations from a model with a random intercept per
## plot; hydrological years labeled by the year in which they end
data(Prec, envir = environment())
data(Temp, envir = environment())
aif <- list(Prec, Temp) |>
    modelFrame(fn = list('moveYr', 'wlai'), lv = list('year', 'year'),
```

```
               form = 'AI ~ year | plot', random = pdDiag(~ 1))
## Mantel correlograms between the two, for each core (100
## permutations to keep the example short)
set.seed(1)
mcomp <- muleMan(trw, aif, rd.var = 'x', nperm = 100)
head(mcomp$mmgram)
```

`mUnits` *Metric system*

Description

This function control metric units.

Usage

```
mUnits(x, from = "mm",
    to = "mm")
```

Arguments

`x`	`numeric` vector.
`from`	`character`. Initial metric unit.
`to`	`character`. Final metric unit.

Details

Characters in `from` and `to` arguments have the form 'p_', where 'p' is the metric prefix and '_' is a base unit. Sixteen metric prefixes are supported: atto 'a', femto 'f', pico 'p', nano 'n', micro 'mm', milli 'm', centi 'c', deci 'd', deca 'da', hecto 'h', kilo 'k', mega 'M', giga 'G', tera 'T', peta 'P', and exa 'E'.

Value

`numeric` vector.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## Simulation of TRW data
set.seed(1)
w <- abs(rnorm(12,1,1))
trw <- ts(w,start = 1970)
## transforming metric units of trw vector from millimeters to meters
sr <- mUnits(trw, from = 'mm', to = 'm')
attributes(sr)
```

`normalizeGroupLevels` *Order grouping levels*

Description

Grouping columns of processed data or of fitted objects are converted to factors whose levels follow numeric order (7, 17, 18, 56) instead of the order of first appearance, or alphabetical order for labels that are not numbers.

Usage

```
normalizeGroupLevels(x,
    groups = NULL, time = NULL,
    keep_time_in_groups = NULL,
    update_meta = TRUE,
    drop_unused = TRUE)
```

Arguments

`x`	A `data.frame`, such as the output of `ringApply`, or a fitted object with a `$fluc` residual series, such as those from `frametoLme`, `modelFrame` or `frametoFriedman`.
`groups`	`character` or `NULL`. Grouping columns to order, as a vector such as `c('plot', 'tree')` or a nested string such as `'plot/tree/sample'`. If `NULL`, the columns are read from the metadata.
`time`	`character` or `NULL`. The active temporal column. If `NULL`, it is read from the metadata.
`keep_time_in_groups`	`logical` or `NULL`. If `TRUE`, the temporal column may stay among the grouping columns; if `FALSE`, it is removed. If `NULL`, the metadata decide, and it is removed by default.
`update_meta`	`logical`. Record the grouping columns in the `BIOdry_meta` attribute of the output.
`drop_unused`	`logical`. Drop unused factor levels.

Details

The grouping columns are those named in the second argument or, if it is not given, those recorded in the `BIOdry_meta` attribute, as set by `defineDataRoles` or `ringApply`. Fitted objects, such as those from `frametoLme` or `modelFrame`, carry no such metadata themselves: the grouping columns are then read from the metadata of their residual series (`$fluc`), else from the grouping formula of the fitted model, and only `$fluc` is ordered; the model itself is unchanged. The active temporal column is not treated as a grouping column unless requested.

A column is ordered as numbers only when all its non-missing labels read as numbers; its levels are then the labels themselves, ordered by their value, so labels such as "07" are kept. Other columns keep their labels, in alphabetical order. Numeric columns become factors with sorted levels. Values and rows are not changed. Ordered levels give a natural panel order in trellis plots.

Value

An object of the same kind as the input: a `data.frame` with ordered grouping columns and its attributes kept, or the fitted object with an ordered `$fluc`. If no data frame is found, the input is returned unchanged with a warning.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## tree-ring widths (Prings05) and radii measured in 2003 (Pradii03);
## Zeide model with ln(time) and a tree-level random part, as in the
## frametoLme example
data(Prings05, envir = environment())
data(Pradii03, envir = environment())
fit <- Prings05 |>
    ringApply(lv = 2, fn = 'rtimes') |>
    ringApply(lv = 'sample', sc.c = Pradii03, rf.t = 2003,
              fn = 'scacum') |>
    frametoLme(form = 'log(x) ~ log(csx) + log(time) | tree',
               random = pdDiag(~ log(csx)))
## tree levels of the residual series, as nlme orders them
levels(fit$fluc$tree)
## in numeric order
fit_n <- normalizeGroupLevels(fit)
levels(fit_n$fluc$tree)
## a data frame, with the grouping columns named
dfm <- Prings05 |>
    ringApply(lv = 2, fn = 'rtimes') |>
    normalizeGroupLevels(groups = 'plot/tree/sample')
lapply(dfm[c('plot', 'tree', 'sample')], levels)
```

Pchron *Tree-ring width (TRW) chronology*

Description

TRW chronology for Pinus pinaster forests from two sample plots in northern Spain (Soria) and east-central Spain (Cuenca). The data set contains eight TRW series. To account for source variability, two trees were selected per site, and two samples were extracted from each tree.

Usage

```
data(Pchron)
```

Format

`data.frame` object with eight columns.

Rows Row names of the data set are time-units labels (years of formation).

Columns Column names represent levels in three within-stand ecological factors that are common to most TRW data and represent variability in stand qualities (stand), tree-radial morphology (sample), and tree genetics/phenotypes (tree). The names are dot-separated labels representing the hierarchy of the three ecological factors, where higher ecological-factor levels are defined first and lower levels after. For example, code 'P16106.17.a' is the column name of sample replicate 'a' in tree '17' in plot 'P16106'.

Details

This data set belongs to a more extensive tree-ring chronology which was processed in previous studies (Bogino and Bravo, 2008) by measuring TRW in polished-core samples (5 mm diameter) using the Windendro program and cross-dating the tree-ring chronologies using COFECHA software. Time units for the TRWs are formation years spanning from 1861 to 2005. The cross-dating records suggested that maritime pine chronologies had high SNR: (23-28 in northern Spain, and 38-61 in east-central Spain) and high EPS (0.96 in northern Spain, and 0.98 in east-central Spain).

References

Bogino, S., and Bravo, F. (2008). Growth response of Pinus pinaster Ait. to climatic variables in central Spanish forests. Ann. For. Sci. 65, 1-13.

Examples

```
str(Pchron)
```

---

plot.frametoFriedman *Plot Friedman fluctuation series over time*

Description

Trellis plots of detrended Friedman fluctuation series are obtained.

Usage

```
## S3 method for class 'frametoFriedman'
plot(x,
    ..., levels = NULL,
    print.trellis = TRUE)
```

Arguments

x	Object inheriting from class frametoFriedman.
...	Further trellis arguments passed to the plotting method.
levels	Optional subset of grouping variables used for conditioning. May be numeric positions or character names.
print.trellis	logical. Print the trellis object.

Details

The default display for frametoFriedman objects is a time-oriented trellis plot of modeled fluctuations or residual-like detrended values over the resolved time variable. This mimics the time-panel displays commonly used with nlme residual objects: the x-axis is the time variable, the y-axis is the detrended fluctuation column, and panels are defined by the declared grouping hierarchy.

When x$fluc inherits from groupedData, its stored formula is honored if it already has the form fluctuation ~ time | groups. If that formula cannot be used safely, the method reconstructs a temporary grouped object with the fluctuation on the left-hand side, the time variable on the x-axis, and the selected grouping variables in the conditioning structure.

Numeric-like grouping variables are normalized to natural numeric order in the plotting copy only, so panel strips such as zones appear in the order 1, 2, 3, ... instead of first appearance.

Value

A trellis plot object. The object is printed by default.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## tree-ring widths (Prings05) and radii measured in 2003 (Pradii03):
## relative time within trees, then cumulative growth scaled to the radii
data(Prings05, envir = environment())
data(Pradii03, envir = environment())
dfm2 <- Prings05 |>
    ringApply(lv = 2, fn = 'rtimes') |>
    ringApply(lv = 'sample', sc.c = Pradii03, rf.t = 2003,
              fn = 'scacum')
ff <- frametoFriedman(dfm2)
## ratio index of each core over time
plot(ff)
## panels by plot, one line per core
plot(ff, levels = 'plot', layout = c(2, 1))
```

`plot.frametoLme` *Plot frametoLme objects*

Description

Trellis plots of modeled fluctuations in `frametoLme` objects are obtained.

Usage

```
## S3 method for class 'frametoLme'
plot(x, ...)
```

Arguments

`x`	An object inheriting from class `frametoLme`.
`...`	Further arguments passed to `plot.nfnGroupedData`, `plot.nmGroupedData` or to trellis plotting methods.

Details

This method makes the fluctuation plot the default visual output of a `frametoLme` object, so users can call `plot()` on the fit directly instead of plotting its `$fluc` component. The method delegates the display to the **nlme** plot methods for grouped data (`plot.nfnGroupedData` and `plot.nmGroupedData`) whenever a grouped fluctuation series is available. Before plotting, it rebuilds a temporary plotting copy in which numeric-like grouping variables are converted to ordered factors with sorted levels, which avoids the common packet order problem where groups such as `3, 1, 2, 5, 6, 7, 4` are displayed in order of appearance rather than natural numeric order.

If `x$fluc` is not available, the method tries to reconstruct a residual fluctuation series from the stored `lme` model and plots Pearson residuals against the recorded time variable. The method also recognizes helper arguments supplied through `...`: `levels` to choose which grouping variables define the conditioning structure, `print.trellis` to print or suppress the trellis object, and any regular trellis or arguments of the **nlme** methods. By default, `as.table = TRUE` is used when the plot is conditioned on groups, which makes the panel order follow a natural table layout.

Value

A trellis object is returned invisibly and printed by default.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## tree-ring widths (Prings05) and radii measured in 2003 (Pradii03);
## Zeide model with ln(time) and a tree-level random part, as in the
## frametoLme example
data(Prings05, envir = environment())
data(Pradii03, envir = environment())
fit <- Prings05 |>
    ringApply(lv = 2, fn = 'rtimes') |>
    ringApply(lv = 'sample', sc.c = Pradii03, rf.t = 2003,
              fn = 'scacum') |>
    frametoLme(form = 'log(x) ~ log(csx) + log(time) | tree',
               random = pdDiag(~ log(csx)))
## Pearson residuals of each tree over time
plot(fit)
## the trellis object can be kept instead of printed
tr <- plot(fit, print.trellis = FALSE)
class(tr)
```

`plot.modelFrame` *Plot modelFrame objects*

Description

Trellis plots of fluctuations in `modelFrame` objects are obtained.

Usage

```
## S3 method for class 'modelFrame'
plot(x, ...)
```

Arguments

`x`	An object inheriting from class `modelFrame`.
`...`	Further trellis arguments passed to the plotting method.

Details

This method is a thin wrapper over the **nlme** plot methods for grouped data (`plot.nfnGroupedData`, `plot.nmGroupedData`) and trellis graphics. For terminal `frametoLme` fits, the default display emulates `plot(model,resid(., type = "p") ~ fitted(.) | GroupID)` by plotting Pearson residuals against fitted values in panels defined by the selected grouping levels. For terminal `frametoFriedman` fits, the default display plots the detrended fluctuation against the smooth fit. For `form = NULL` outputs, the method plots the processed fluctuation against time.

All regular trellis arguments are accepted through `....` The method recognizes three additional helper arguments in `...`: `levels` to choose which grouping variables define the conditioning panels, `reference` to control the horizontal reference line, and `print.trellis` to print or suppress the trellis object.

Value

A trellis plot object. The object is printed by default.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## ring-width chronologies of eight cores and the allometric
## parameters of the modelFrame example
data(Pchron, envir = environment())
biom_param <- c(2.87, 0.85, 0.05, 2.5)
## the biomass model on year with a random intercept per tree
trwf <- Pchron |>
    modelFrame(to = 'cm', MoreArgs = list(mp = c(2, 1, biom_param)),
               form = 'log(x) ~ log(csx) + year | tree',
               random = pdDiag(~ 1))
## Pearson residuals against fitted values for each tree
plot(trwf)
## the four trees side by side
plot(trwf, layout = c(4, 1))
## with form = NULL, the processed series are plotted over time
Pchron |>
    modelFrame(form = NULL, to = 'cm',
               MoreArgs = list(mp = c(2, 1, biom_param))) |>
    plot()
```

---

plot.muleMan *Plot muleMan objects*

Description

Diagnostic Trellis plot for fluctuations in `muleMan` objects are obtained.

Usage

```
## S3 method for class 'muleMan'
plot(x, ...)
```

Arguments

x	An object inheriting from class `muleMan`.
...	further arguments passed to the Trellis plot function.

Details

One panel is drawn for each correlogram. Panels are ordered by the grouping levels from the outermost to the innermost (plot, then tree, then core in the bundled data); numeric labels are ordered as numbers. Filled points mark lags with p-values below 0.05. The axes are labeled "Lag" and "Mantel r" unless `xlab` or `ylab` are given.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## ring-width chronologies (mm) of eight cores and the allometric
## parameters of the modelFrame example
data(Pchron, envir = environment())
biom_param <- c(2.87, 0.85, 0.05, 2.5)
## tree-biomass series of each core, processed without a terminal
## model (on these 2 plots x 2 trees x 2 cores no mixed model with plot,
## tree and core levels is well posed), with the columns muleMan pairs
## on: the series, year and the plot, tree and core labels
trw <- Pchron |>
    modelFrame(form = NULL, to = 'cm',
               MoreArgs = list(mp = c(2, 1, biom_param)))
trw <- trw[, c('x', 'year', 'plot', 'tree', 'sample')]
## aridity-index fluctuations from a model with a random intercept per
## plot; hydrological years labeled by the year in which they end
data(Prec, envir = environment())
data(Temp, envir = environment())
aif <- list(Prec, Temp) |>
    modelFrame(fn = list('moveYr', 'wlai'), lv = list('year', 'year'),
               form = 'AI ~ year | plot', random = pdDiag(~ 1))
## Mantel correlograms between the two, for each core (100
## permutations to keep the example short)
set.seed(1)
mcomp <- muleMan(trw, aif, rd.var = 'x', nperm = 100)
## one correlogram per core; filled points mark significant lags
plot(mcomp)
```

`plot.wlai` *Plot Walter-Lieth climate diagrams*

Description

A Walter-Lieth climate diagram of one annual block of a `wlai` object is drawn, with its aridity index.

Usage

```
## S3 method for class 'wlai'
plot(x, y = NULL,
    which = NULL, subset = NULL,
    year = NULL, hydro_year_id = NULL,
    hatch = TRUE, dry_density = 18,
    moist_density = 18,
    dry_angle = 45, moist_angle = -45,
    ...)
```

Arguments

x	A wlai object.
y	Not used; kept for the plot() generic.
which	numeric position or character name of the block to draw.
subset	Named list (or one-row data.frame) of column values of the wlai table selecting the block, such as the plot label and a period label.
year	Label of the year column of the block.
hydro_year_id	Label of the hydro_year_id column of the block.
hatch	logical. Hatch the dry and moist seasons; if FALSE, fill them in gray.
dry_density	numeric. Hatching density of the dry season, in lines per inch.
moist_density	numeric. Hatching density of the moist season, in lines per inch.
dry_angle	numeric. Hatching angle of the dry season, in degrees.
moist_angle	numeric. Hatching angle of the moist season, in degrees.
...	Further graphical parameters passed to plot.default.

Details

Precipitation (right axis, mm) and temperature (left axis, degrees C) are drawn on the Walter-Lieth scale computed by wlai: 10 degrees C match 20 mm (P = 2T), and precipitation above 100 mm is compressed. Months whose precipitation curve lies below the temperature curve form the dry season; the others form the moist season, and precipitation above 100 mm is filled in blue. The aridity index (AI) is printed above the diagram, followed by the grouping labels and the year label of the block; that label follows the labeling chosen in moveYr and wlai: P44005.2001 for October 2000 to September 2001 with the default 'end'.

A wlai object computed on long-format data (several plots or years) holds one annual block per plot and period. Block names join the grouping labels and the hydro_year_id of the block, the calendar label of its first month: P44005.2000 holds October 2000 to September 2001. A block is chosen by position or block name, by the labels of the year or hydro_year_id columns of the wlai table, or by a named list of column values. Without a selection the first block is drawn, with a warning; a selection that matches several blocks stops with an error.

Value

Called for its side effect, the diagram; NULL is returned invisibly.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## monthly precipitation and temperature of plot P44005 (PTclim05);
## hydrological years from October to September, labeled by the
## calendar year in which they end
data(PTclim05, envir = environment())
ai <- subset(PTclim05, plot == 'P44005') |>
    moveYr() |>
    wlai()
subset(ai, year == 2001)
## the diagram of October 2000 to September 2001
plot(ai, year = 2001)
## the same block, with gray fills
plot(ai, hydro_year_id = 2000, hatch = FALSE)
```

Pradii03 *Vector of radii measured on P.pinaster*

Description

Vector of radii measured in dominant trees of P.pinaster (2003) growing on the Ebro river basin, Spain

Usage

```
data(Pradii03)
```

Format

A numeric vector with names of the vector indicating the measured tree.

Details

This vector contains inside-bark radii, measured in four trees during 2003 (mm). This radial vector was derived from the over-bark diameters at breast height (cm), with allometric estimation of inside-bark diameters from the over-bark diameters. Names of the vector correspond to labels at tree level

Source

Sustainable Forest Management Research Institute, Universidad de Valladolid (UVa)-INIA.

Examples

```
str(Pradii03)
```

Prec *Cumulative precipitations*

Description

Monthly cumulative precipitation (mm).

Usage

```
data(Prec)
```

Format

`data.frame` object with 24 records.

Column Column names contain plot codes and monthly abbreviations

Details

Recorded months begin at January and end at December, and observed years span from 1951 to 2005. Factor in data set defines one sample level (plot)

References

Bogino, S., and Bravo, F. (2008). Growth response of Pinus pinaster Ait. to climatic variables in central Spanish forests. Ann. For. Sci. 65, 1-13.

Examples

```
str(Prec)
```

Prings05 *Multilevel data set of P. pinaster*

Description

Radial increments of Pinus pinaster from two sample plots located on Northern and Southern portions of Ebro river basin, Spain.

Usage

```
data(Prings05)
```

Format

A data frame with the following 5 variables.

`x` A numeric vector with the radial increments in mm year-1

`year` A numeric vector with the recorded year

`sample` A factor indicating the sample replicate

`tree` A factor indicating the tree number

`plot` A factor indicating the plot code

Details

This data set contains eight series of tree-ring widths of maritime pine (Pinus pinaster), with recorded years spanning from 1810 to 2005. The cores were sampled from dominant trees of two sites, with sample plots being located on both: northern Spain (plot code: P44005) and center-east portion of the same country (plot code: P16106). Two trees were selected by plot, and two core samples were extracted by tree. Consequently, the sample design defined three levels: sample in tree on plot (plot level), sample in tree (tree level), and sample level.

References

Bogino, S., and Bravo, F. (2008). Growth response of Pinus pinaster Ait. to climatic variables in central Spanish forests. Ann. For. Sci. 65, 1-13.

Examples

```
str(Prings05)
```

`print.frametoFriedman` *Print frametoFriedman objects*

Description

A compact printed representation of a `frametoFriedman` object is obtained.

Usage

```
## S3 method for class 'frametoFriedman'
print(x,
    ...)
```

Arguments

`x`	Object inheriting from class `"frametoFriedman"`.
`...`	Additional arguments passed to `summary.frametoFriedman`.

Details

This method prints the compact summary instead of the raw list returned by `frametoFriedman`.

Value

The compact printed representation.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## tree-ring widths (Prings05) and radii measured in 2003 (Pradii03):
## relative time within trees, then cumulative growth scaled to the radii
data(Prings05, envir = environment())
data(Pradii03, envir = environment())
dfm2 <- Prings05 |>
    ringApply(lv = 2, fn = 'rtimes') |>
    ringApply(lv = 'sample', sc.c = Pradii03, rf.t = 2003,
              fn = 'scacum')
## printing shows the compact summary, not the nested list
dfm2 |> frametoFriedman() |> print()
```

---

print.frametoLme *Print frametoLme objects*

Description

A compact printed representation of a `frametoLme` object is obtained.

Usage

```
## S3 method for class 'frametoLme'
print(x, ...)
```

Arguments

`x`	Object inheriting from class `"frametoLme"`.
`...`	Additional arguments passed to `summary.frametoLme`.

Details

This method prints the compact summary instead of the raw list returned by `frametoLme`, so users do not see the full nested list with model, call, and residual series every time the object name is evaluated.

Value

The compact printed representation.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## tree-ring widths (Prings05) and radii measured in 2003 (Pradii03):
## relative time within trees, then cumulative growth scaled to the radii
data(Prings05, envir = environment())
data(Pradii03, envir = environment())
dfm2 <- Prings05 |>
    ringApply(lv = 2, fn = 'rtimes') |>
    ringApply(lv = 'sample', sc.c = Pradii03, rf.t = 2003,
              fn = 'scacum')
## printing shows the compact summary, not the nested list
dfm2 |>
    frametoLme(form = 'log(x) ~ log(csx) + log(time) | tree',
               random = pdDiag(~ log(csx))) |>
    print()
```

---

print.modelFrame *Print modelFrame objects*

Description

A compact printed representation of a modelFrame object is obtained.

Usage

```
## S3 method for class 'modelFrame'
print(x, ...)
```

Arguments

x	Object inheriting from class modelFrame.
...	Additional arguments passed to summary.modelFrame.

Value

The compact printed representation.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
data(Pchron, envir = environment())
biom_param <- c(2.87, 0.85, 0.05, 2.5)
## printing shows a compact description, not the whole data
Pchron |>
    modelFrame(form = NULL, to = 'cm',
               MoreArgs = list(mp = c(2, 1, biom_param))) |>
    print()
```

`print.summary.frametoFriedman` *Print compact frametoFriedman summaries*

Description

A compact printed summary of a `frametoFriedman` object is obtained.

Usage

```
## S3 method for class 'summary.frametoFriedman'
print(x,
    digits = max(3L,
        getOption("digits") -
            2L), ...)
```

Arguments

`x`	Object inheriting from class `"summary.frametoFriedman"`.
`digits`	Integer number of `digits`.
`...`	Unused extra arguments.

Value

The printed compact summary.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## tree-ring widths (Prings05) and radii measured in 2003 (Pradii03):
## relative time within trees, then cumulative growth scaled to the radii
data(Prings05, envir = environment())
data(Pradii03, envir = environment())
dfm2 <- Prings05 |>
    ringApply(lv = 2, fn = 'rtimes') |>
```

```
    ringApply(lv = 'sample', sc.c = Pradii03, rf.t = 2003,
              fn = 'scacum')
dfm2 |> frametoFriedman() |> summary() |> print()
```

`print.summary.frametoLme` *Print compact frametoLme summaries*

Description

A compact printed summary of a `frametoLme` object is obtained.

Usage

```
## S3 method for class 'summary.frametoLme'
print(x,
    digits = max(3L,
        getOption("digits") -
            2L), ...)
```

Arguments

`x`	Object inheriting from class `"summary.frametoLme"`.
`digits`	Integer number of `digits`.
`...`	Unused extra arguments.

Value

The printed compact summary.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## tree-ring widths (Prings05) and radii measured in 2003 (Pradii03):
## relative time within trees, then cumulative growth scaled to the radii
data(Prings05, envir = environment())
data(Pradii03, envir = environment())
dfm2 <- Prings05 |>
    ringApply(lv = 2, fn = 'rtimes') |>
    ringApply(lv = 'sample', sc.c = Pradii03, rf.t = 2003,
              fn = 'scacum')
dfm2 |>
    frametoLme(form = 'log(x) ~ log(csx) + log(time) | tree',
               random = pdDiag(~ log(csx))) |>
    summary() |>
    print(digits = 3)
```

`print.summary.modelFrame` *Print compact modelFrame summaries*

Description

A compact printed summary of a `modelFrame` object is obtained.

Usage

```
## S3 method for class 'summary.modelFrame'
print(x,
    digits = max(3L,
        getOption("digits") -
            2L), ...)
```

Arguments

`x`	Object inheriting from class `"summary.modelFrame"`.
`digits`	Integer number of `digits`.
`...`	Unused extra arguments.

Value

The printed compact summary.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
data(Pchron, envir = environment())
biom_param <- c(2.87, 0.85, 0.05, 2.5)
Pchron |>
    modelFrame(to = 'cm', MoreArgs = list(mp = c(2, 1, biom_param)),
               form = 'log(x) ~ log(csx) + year | tree',
               random = pdDiag(~ 1)) |>
    summary() |>
    print(digits = 3)
```

PTclim05 *Multilevel data set of monthly cumulative precipitations and monthly average temperatures*

Description

Monthly precipitation sums and average temperatures of two climatic locations.

Usage

```
data(PTclim05)
```

Format

A data frame with the following 5 variables.

`pr` A numeric vector with the monthly cumulative precipitations (mm month-1)

`tm` A numeric vector with the monthly average temperatures (degree C month-1)

`month` A numeric vector of months

`year` A numeric vector of years

`plot` A factor vector of plot codes

Details

Recorded months begin at January and end at December, and observed years span from 1951 to 2005. Factor in data set defines one sample level (plot)

References

Bogino, S., and Bravo, F. (2008). Growth response of Pinus pinaster Ait. to climatic variables in central Spanish forests. Ann. For. Sci. 65, 1-13.

Examples

```
str(PTclim05)
```

ringApply *Multilevel apply*

Description

Wrapper of Map to apply functions on multilevel data series and preserve factor-level structure in the outputs.

Usage

```
ringApply(rd, lv = 1,
    fn = "scacum", focus = NULL,
    target = NULL, focus_mode = c("auto",
        "vector", "args",
        "data.frame"),
    validate = c("none",
        "warn", "stop"),
    ...)
```

Arguments

rd	data.frame. Multilevel ecological data series.
lv	numeric position, or character name, of an ecological factor in the processed MEDS.
fn	character name of the function to be evaluated (see details). Default 'scacum' computes scaled-cumulative radii.
focus	Optional focus column or columns. Use NULL to keep the historical full-data dispatch for core BIOdry helpers and otherwise default to the active response stored in BIOdry_meta. Use ".data" to force full-data-frame dispatch.
target	Optional output column name used when a focused helper returns a single vector of the same length as the split block. By default the focused column is replaced when a single focus column is used.
focus_mode	focus_mode
validate	Validation mode. Use "none" for quiet backward-compatible behavior, "warn" for diagnostics, or "stop" for strict checking.
...	Further arguments in the function being specified fn argument (see details)

Details

Other functions such as rtimes, scacum, amod, or wlai can be implemented. Function arguments should be formulated as suggested in mapply, with constant arguments being stored in a MoreArgs list. This implementation keeps MoreArgs constant across split blocks. The split levels are the factor and character columns other than the month column (see cClass); whole-number columns such as year are split levels only when requested through lv. The nesting structure is resolved through slitFrame, which is the infrastructure helper used to materialize the hierarchy defined by

lv. When lv is supplied, whether as a numeric position or as a character level name, that choice controls the split hierarchy explicitly.

After splitting and before applying fn, each block is ordered on the time variables that remain inside the block. Variables used as split levels are excluded from the ordering keys. Thus a call such as lv = 'year' orders within yearly blocks by finer time variables such as month, day, or relative time rather than by year again, whereas a call such as lv = 'sample' orders within each ring series by year and then by any finer time variable. Ordering prefers hydro_year_id, then year-like variables not used for splitting, then month-like variables, and finally relative time variables such as time, age, or cambial_age. This contract is important because other BIOdry functions can rely on ringApply to deliver already split and time-ordered blocks. Split variables used internally for a transformation step are recorded in the metadata history, but they are not promoted automatically to the persistent grouping hierarchy used later by frametoLme or frametoFriedman. In particular, a derived time key such as hydro_year_id may be the active time variable after moveYr while remaining outside the grouping hierarchy unless the user explicitly requested otherwise in defineDataRoles.

Duplicated output column names are collapsed when they are exact duplicates. Output columns are ordered explicitly with record or measured variables first (for example x, csx, pr, pm, or tm), then time variables (for example hour, day, week, month, or year), and finally grouping factors ordered from the lowest available hierarchy to the broadest hierarchy (for example sample before tree, plot, zone, or biome). By default, validation is silent for backwards compatibility; use validate = "warn" or validate = "stop" for stricter diagnostics.

This patched version also attaches a small workflow attribute named "BIOdry_meta" to the returned data frame. The attribute stores lightweight context from the current evaluation, such as the applied function name, the split levels, a compact summary of relevant arguments, simple role hints (response, time, grouping variables), and a conservative variable-name map when a known transformation creates a new key, for example year -> hydro_year_id after moveYr. The role hints are taken from column types and record context only: they do not override the type rule in lmeForm, tdForm, frametoLme or FriedmanForm; only roles declared with defineDataRoles do. The attribute does not alter the returned columns.

When constructor metadata or prior

ringApply steps indicate that a time variable has been replaced by a derived annual key, character values of

lv are resolved against that current key before splitting. Thus a workflow such as

moveYr -> wlai can keep using

lv = 'year' while internally splitting on hydro_year_id when that derived variable has become the active annual index. The original calendar year column is retained for reference when the applied function returns it.

For climatic long-format data, a safer workflow is often to first use moveYr to create a stable annual key such as hydro_year_id, then split by the relevant factor columns plus that year key.

Value

data.frame object preserving initial factor-level columns.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

References

Lara, W., F. Bravo, D. Maguire. 2013. Modeling patterns between drought and tree biomass growth from dendrochronological data: A multilevel approach. Agric. For. Meteorol., 178-179:140-151.

Examples

```
##Multilevel ecological data series (MEDS) of tree-ring widths:
data(Prings05,envir = environment())
## Radial increments measured on 2003:
data(Pradii03,envir = environment())
## MEDS of monthly precipitation sums and average temperatures:
data(PTclim05,envir = environment())

##Tree-level scaling of years of formation
##with 'rtimes' function:
dfm1 <- ringApply(Prings05,
                  lv = 2,
                  fn = 'rtimes')
str(dfm1)
##Relative time-units from year 1 to year 9:
subset(dfm1,time%in%c(1:9,NA))

## Sample-level scaling of TRW chronologies around reference radii
## which were measured at 2003:
dfm2 <- ringApply(dfm1,
                  lv = 'sample',
                  sc.c = Pradii03,
                  rf.t = 2003,
                  fn = 'scacum')
str(dfm2)
##Sample-level modeling of basal areas (mm2) via allometric
##scaling:
dfm3 <- ringApply(dfm2,
                  lv = 'sample',
                  fn = 'amod',
                  MoreArgs = list(mp = c(2,1,0.25 * pi,2)))
str(dfm3)

## Seasonal years from 'October' to 'September':
cl1 <- ringApply(PTclim05,
                 lv = 'year',
                 fn = 'moveYr')
tail(cl1,15)
attr(cl1, 'BIOdry_meta')

##Year-level aridity indexes:
wl <- ringApply(cl1,
                lv = 'year',
                fn = 'wlai')
str(wl)
attr(wl, 'BIOdry_meta')
```

```
## Sample-level unit conversion on the active response:
xu <- ringApply(Prings05,
                lv = 'sample',
                fn = 'mUnits',
                from = 'mm',
                to = 'cm')
head(xu)

## Plot of aridity-index fluctuations:
d <- groupedData(lmeForm(wl),wl)
plot(d)
```

`rtimes` *Time-units synchronization*

Description

Unique observations in time-series replicates are excluded

Usage

```
rtimes(x, only.dup = TRUE,
    lv = NULL, validate = c("none",
        "warn", "stop"))
```

Arguments

`x`	multilevel ecological data series containing a column of time units, or `numeric` vector with names representing the time units. In a data frame the time units are found by the package's column names: `year` (or `hydro_year_id` from `moveYr`), `month` and `time`; a date column is also used. Without any of them, row order is used and a warning is given.
`only.dup`	`logical`. Extract only duplicated times. If TRUE then unique times are replaced with NA. If all computed times are unique then this argument is ignored. Rows with NA time, for example the first years of a core that no other core of its tree covers (16 rows of `Prings05` with `lv = 2` in `ringApply`), are later dropped from any model that uses `time` (see `frametoLme`).
`lv`	optional split level passed to `slitFrame()`. When supplied, synchronized times are computed within each split block rather than on the whole table.
`validate`	validation mode.

Value

`data.frame` object with the initial vector and its time units.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## row names of a vector
fy <- function(y,span){(y - span):y}
x <- c(fy(2005,5),fy(2007,10))
## (not run) Simulating the vector
r <- abs(rnorm(length(x)))
names(r) <- x
## (not run) computing the synchronized times:
rtimes(r,only.dup = TRUE)
## (not run) Extracting only duplicated times:
na.omit(rtimes(r,only.dup = TRUE))
```

---

scacum *scacum*

Description

This function computes cumulative and optionally scaled sums of ordered time-series replicates.

Usage

```
scacum(x, sc.c = NA,
    rf.t = NA, lv = NULL,
    validate = c("none",
        "warn", "stop"))
```

Arguments

x	Numeric vector of time-series replicates, or a data frame.
sc.c	Numeric scaling constant. If NA, cumulative sums are not scaled.
rf.t	Reference time for scaling. If NA, the last time value of each ordered block is used. If given, the data must have a year, hydro_year_id or time column; otherwise scacum() stops.
lv	Optional split level for multilevel data frames. Can be NULL, character, or numeric, and is resolved through slitFrame().
validate	Validation mode.

Details

Cumulative sums of time-series replicates (e.g. tree-ring widths) can be computed on a single ordered block, or within multilevel data frames after splitting by a specified level. For data frames, `scacum()` orders each block on available time variables before computing cumulative sums and returns the original data with an added `csx` column. The value column is `x`, the package's column name; a response declared with `defineDataRoles` takes precedence. Without either, `scacum()` stops. The time variables are found by the package's column names: `year` (or `hydro_year_id` from `moveYr`), `month` and `time`; `rf.t` refers to the first of `hydro_year_id`, `year` and `time` that is present.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
x <- c(0.79,0.32,0.53,0.43,0.18)
names(x) <- 1948:1952
scacum(x, sc.c = 4, rf.t = 1951)

max(cumsum(x))
scacum(x, NA, 1951)
```

`shiftFrame` *shiftFrame*

Description

dendroclimatic chronologies (trw, and climatic data) are formatted into multilevel ecological data series. SI units of continuous variables in the data can be transformed.

Usage

```
shiftFrame(rd, f.nm = NULL,
    x.nm = names(rd)[1L],
    t.nm = "year", split_pattern = "\\.",
    level_sep = ".",
    ...)
```

Arguments

`rd`	rd
`f.nm`	f.nm
`x.nm`	x.nm
`t.nm`	t.nm

split_pattern	split_pattern
level_sep	level_sep
...	...

Details

Wide BIOdry data may represent either tree-ring or climatic series. The patched implementation prefers explicit time columns in wide data, that is, the column named by t.nm, whole-number numeric columns and the month column, and then treats the remaining value columns as hierarchy-encoded series labels. This rule applies to both climate and tree-ring wide data. Hierarchy levels are parsed from value-column names with the configurable regular expression given by split_pattern. The default keeps the historical dot-separated BIOdry convention.

When long data are shifted to wide format, time variables are retained as independent columns rather than being hidden in row names. Value columns are then rebuilt from the grouping hierarchy using level_sep. When long data contain more than one measured variable, for example tm and pm, the wide columns are prefixed by the measure name, such as tm.z1 or pm.z1, so the reverse conversion can recover all value variables. This makes wide to long round-trips safer for data with more than one time variable, such as year + month in climate series or year + cambial_age in tree-ring series.

Row-name time is still accepted as a backward- compatibility fallback for older BIOdry wide chronologies. In that case, row names are used only when explicit time columns are absent.

When a wide object produced by shiftFrame() is shifted back to long format, the stored metadata for x.nm, f.nm, t.nm, and time_cols take precedence over the type rule (see cClass). This makes wide-to-long round-trips deterministic. For single-response long results that are returned as groupedData, the grouping formula is built from the actual response variable and the stored hierarchy, rather than being derived again from column types. Long-format outputs are ordered explicitly as record or measured variables first, then time variables, and finally grouping factors from the lowest resolved hierarchy level to the broadest one.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
data(Prings05, envir = environment())
pwide <- shiftFrame(Prings05, from = 'mm', to = 'mmm')
str(pwide)
plong <- shiftFrame(pwide, from = 'mmm', to = 'mm')
plot(plong)
```

slitFrame *slitFrame*

Description

Robust multilevel splitting

Usage

```
slitFrame(rd, lv = NULL,
    validate = c("none",
        "warn", "stop"))
```

Arguments

rd	data.frame with grouping columns.
lv	NULL, character, or numeric selector for splitting columns.
validate	validation mode.

Value

list of data.frames.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
data(Prings05, envir = environment())
data(PTclim05, envir = environment())
spl <- slitFrame(Prings05)
str(spl)
spl <- slitFrame(Prings05, "year")
str(spl)
spl <- slitFrame(PTclim05, c("plot", "year"))
str(spl)
```

summary.frametoFriedman *Summarize frametoFriedman objects*

Description

A compact summary of a `frametoFriedman` `object` is obtained.

Usage

```
## S3 method for class 'frametoFriedman'
summary(object,
    ...)
```

Arguments

`object`	Object inheriting from class `"frametoFriedman"`.
`...`	Unused extra arguments.

Details

This method extracts the main components that users typically inspect after Friedman detrending: the original call, the chosen response and time variables, the series hierarchy, the smoothing settings, the number of observations and series, and a per-series status table.

Value

A compact summary `object` of class `"summary.frametoFriedman"`.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## tree-ring widths (Prings05) and radii measured in 2003 (Pradii03):
## relative time within trees, then cumulative growth scaled to the radii
data(Prings05, envir = environment())
data(Pradii03, envir = environment())
dfm2 <- Prings05 |>
    ringApply(lv = 2, fn = 'rtimes') |>
    ringApply(lv = 'sample', sc.c = Pradii03, rf.t = 2003,
              fn = 'scacum')
s <- dfm2 |> frametoFriedman() |> summary()
## settings and the status of each series
s$span
s$series
```

summary.frametoLme *Summarize frametoLme objects*

Description

A compact summary of a frametoLme object is obtained.

Usage

```
## S3 method for class 'frametoLme'
summary(object,
    ...)
```

Arguments

object	Object inheriting from class "frametoLme".
...	Additional arguments passed to summary.lme.

Details

This method extracts the main model components that users typically inspect after fitting a mixed-effects model: the original package call, fitting method, sample size, likelihood-based fit statistics, residual standard deviation, grouping sizes, fixed-effect table, and random-effect variance structure. It avoids the very long Data line (a deparsed structure() call) shown by summary.lme when the fitted lme object stores an inlined groupedData call.

Value

A compact summary object of class "summary.frametoLme".

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## tree-ring widths (Prings05) and radii measured in 2003 (Pradii03):
## relative time within trees, then cumulative growth scaled to the radii
data(Prings05, envir = environment())
data(Pradii03, envir = environment())
dfm2 <- Prings05 |>
    ringApply(lv = 2, fn = 'rtimes') |>
    ringApply(lv = 'sample', sc.c = Pradii03, rf.t = 2003,
              fn = 'scacum')
s <- dfm2 |>
    frametoLme(form = 'log(x) ~ log(csx) + log(time) | tree',
               random = pdDiag(~ log(csx))) |>
```

```
    summary()
## fit statistics and fixed effects
s$AIC
s$fixed
```

`summary.modelFrame` *Summarize modelFrame objects*

Description

A compact summary of a `modelFrame` `object` is obtained.

Usage

```
## S3 method for class 'modelFrame'
summary(object,
    ...)
```

Arguments

`object`	Object inheriting from class `modelFrame`.
`...`	Additional arguments passed to the terminal summary method.

Details

This method wraps the terminal fitter stored in the `modelFrame` `object`. If `form` = `NULL`, the method returns a compact description of the processed long-format data and its grouped plotting structure.

Value

A compact summary `object` for `modelFrame`.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
data(Pchron, envir = environment())
biom_param <- c(2.87, 0.85, 0.05, 2.5)
## the biomass model on year with a random intercept per tree
s <- Pchron |>
    modelFrame(to = 'cm', MoreArgs = list(mp = c(2, 1, biom_param)),
               form = 'log(x) ~ log(csx) + year | tree',
               random = pdDiag(~ 1)) |>
    summary()
## fit statistics and fixed effects of the terminal model
s$AIC
```

```
s$fixed
## with form = NULL the summary describes the processed data
Pchron |>
    modelFrame(form = NULL, to = 'cm',
               MoreArgs = list(mp = c(2, 1, biom_param))) |>
    summary()
```

---

tdForm *ltd formulas*

Description

This function formulates linear time-decline formulas (ltd) from categorical variables in multilevel ecological data series.

Usage

```
tdForm(rd, prim.cov = FALSE,
    on.time = TRUE, log.t = FALSE,
    lev.rm = NULL, validate = c("none",
        "warn", "stop"))
```

Arguments

rd	data.frame or character vector. Multilevel ecological data series or vector of ecological factors.
prim.cov	logical. Print a primary covariate form: '~ cov'. If FALSE then a complete formula: 'resp ~ cov | group' is printed.
on.time	logical. If TRUE (default) the relative time column 'time' from rtimes is used when present, otherwise 'year'. If FALSE 'year' is used when present, otherwise 'time'. A time variable declared with defineDataRoles(time = ...) takes precedence over both.
log.t	logical. If TRUE then f(time) = ln(time). Default FALSE produces a log-linear time-decline formula. Both settings are Zeide forms, f(time) = time (FALSE) and f(time) = ln(time) (TRUE), with the same number of parameters; fit both by maximum likelihood (method = "ML" in frametoLme) and choose by AIC or BIC.
lev.rm	NULL or character name of the ecological factor(s) in the MEDS to be removed from the formula.
validate	character. Validation mode: 'none' silences checks, 'warn' emits warnings, and 'stop' stops on invalid inputs.

Details

the ltd formulas belong to following general equation: log (x) = log (csx) + f(time); where the relative organic growth (x) is explained by the cumulative organic growth (csx) plus a function of time f(time); with f(time) being either the time or a logarithmic transformation the time. The ltd can be implemented by `modelFrame` function to subtract trends in organic MEDS.

The response `x`, the cumulative growth `csx` and the time covariate (`time` from `rtimes`, or `year`) are the package's fixed column names. The grouping levels are the factor and character columns other than the month column, outer level first (the reverse of the column order; see `cClass`). A time variable or grouping levels declared with `defineDataRoles` take precedence; the metadata that `ringApply` records automatically does not.

Value

`formula` with the forms: 'resp ~ cov | group' or '~ cov'.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

References

Zeide B. 1993. Analysis of Growth Equations. For. Sci., 39: 594-616.

Examples

```
## an ltd formula:
lev <- c('plot','tree')
tdeq <- tdForm(lev,log.t = TRUE)
tdeq
## (not run) only primary covariate:
tdeq1 <- tdForm(lev,prim.cov = TRUE)
tdeq1
##Multilevel data frame of tree-ring widths:
data(Prings05,envir = environment())
## removing two levels: 'plot' and 'tree' from the formula
tdea2 <- tdForm(Prings05, lev.rm = c('plot','tree'))
tdea2 <- tdForm(Prings05, lev.rm = 2:3)
```

Temp *Average temperatures*

Description

Monthly average temperatures (degree C).

Usage

```
data(Temp)
```

Format

`data.frame` object with 24 records.

Column Column names contain plot codes and monthly abbreviations

Details

Recorded months begin at January and end at December, and observed years span from 1951 to 2005. Factor in data set defines one sample level (plot)

References

Bogino, S., and Bravo, F. (2008). Growth response of Pinus pinaster Ait. to climatic variables in central Spanish forests. Ann. For. Sci. 65, 1-13.

Examples

```
str(Temp)
```

---

update.frametoLme *Update frametoLme objects*

Description

A `frametoLme` fit is refitted after one or more arguments of its original call are changed.

Usage

```
## S3 method for class 'frametoLme'
update(object,
    ..., evaluate = TRUE)
```

Arguments

`object`	A `frametoLme` fit.
`...`	Named arguments replacing or adding entries in the original `frametoLme()` call, such as `method = 'ML'` or a new formula string.
`evaluate`	`logical`. Refit the model. If `FALSE`, the rebuilt call is returned instead, with the stored model data written as `..rd..`

Details

The original `frametoLme()` call is rebuilt with the new arguments and evaluated on the data stored with the fitted model (`$model$data`), not on the data named in the call. Refitting therefore works when the original call came from a pipeline, and the grouping columns, metadata and time columns created upstream remain available. Arguments not given keep their original values. Any argument of `frametoLme()` can be changed, including those it passes to `lmeForm`, `tdForm` or `lme`, such as `method`, `random` or `lev.rm`. The refitted model keeps the data expression of the original call, so it prints like the original. Fits compared with `anova.frametoLme` must use the same data, and models that differ in their fixed effects must be fitted by maximum likelihood.

Value

A refitted `frametoLme` fit, or the rebuilt call.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

References

Pinheiro J. C., D. M. Bates. 2000. Mixed-effects models in S and S-PLUS. Springer, New York.

Examples

```
## tree-ring widths (Prings05) and radii measured in 2003 (Pradii03);
## Zeide model with ln(time) and a tree-level random part, as in the
## frametoLme example
data(Prings05, envir = environment())
data(Pradii03, envir = environment())
fit <- Prings05 |>
    ringApply(lv = 2, fn = 'rtimes') |>
    ringApply(lv = 'sample', sc.c = Pradii03, rf.t = 2003,
              fn = 'scacum') |>
    frametoLme(form = 'log(x) ~ log(csx) + log(time) | tree',
               random = pdDiag(~ log(csx)))
## the same model fitted by maximum likelihood
fit_lnt <- update(fit, method = 'ML')
fixef(fit_lnt$model)
## the other Zeide time form, f(t) = t, also by maximum likelihood
fit_t <- update(fit_lnt, form = 'log(x) ~ log(csx) + time | tree')
c(ln_t = AIC(fit_lnt$model), t = AIC(fit_t$model))
```

---

wlai *Walter-Lieth aridity index*

Description

Computing the annual aridity index from Walter-Lieth climate diagrams

Usage

```
wlai(cd, sqt = TRUE,
    ini.mnt = NULL, label_year = c("end",
        "start"), validate = c("none",
        "warn", "stop"))
```

Arguments

cd	data.frame. Multilevel climatic data series of monthly precipitation sums (mm), and monthly average temperatures (degree C). Precipitation is the column pr and temperature the column tm; inputs declared with defineDataRoles take precedence: declared pr and tm are used by name, other declared inputs in their order, precipitation first. If pr or tm is missing, the first two numeric columns are used and a warning names them. The month column is found by the package's column name month (a factor whose levels are month names is also accepted), and the year column by year, or hydro_year_id as returned by moveYr; without them a single whole-number column is used as the year. If none of year, hydro_year_id, month or time is present (and the row names are not month names), a warning is given.
sqt	logical. Print the square root of the aridity index.
ini.mnt	character or numeric. Optional start month to define annual blocks in long climatic data. When supplied, yearly labels are shifted before computing aridity indexes. Without ini.mnt, calendar blocks start at the first month of the earliest year; if the data have no year column, the month of the first row is used, so rows should then be in time order.
label_year	character. How hydrological years are labeled in the output column year. 'end' (default) labels each year by the calendar year in which it ends, so October 2000 to September 2001 is labeled 2001, the year of ring formation; 'start' labels it by the calendar year in which it starts. It applies when ini.mnt is supplied or when cd already has a hydro_year_id column, as returned by moveYr. The output column hydro_year_id always holds the start year. If the year column of cd follows the other convention, a warning is given and label_year decides. Without hydrological years the years are calendar years and label_year has no effect.
validate	character. Validation mode.

Value

A data.frame of class wlai with one row per annual block: the aridity index AI, the year columns (hydro_year_id and year when the input comes from moveYr) and the grouping columns. The Walter-Lieth geometry of each block is kept in attributes and drawn by plot.wlai; no plot is drawn here.

Author(s)

Wilson Lara-Henao [aut, cre] (ORCID: <https://orcid.org/0000-0003-3527-1380>), Felipe Bravo [aut] (ORCID: <https://orcid.org/0000-0001-7348-6695>)

Examples

```
## monthly precipitation (pr, mm) and temperature (tm, degrees C) of
## two plots, 1955-2005 (PTclim05)
data(PTclim05, envir = environment())
## aridity index of one calendar year of one plot
subset(PTclim05, plot == 'P44005' & year == 2000) |> wlai()
## one index per plot and hydrological year (October to September),
## labeled by the year in which it ends; the first and last
## hydrological years are incomplete and get NA
ai <- PTclim05 |>
    moveYr() |>
    wlai()
head(ai)
subset(ai, is.na(AI))
## the Walter-Lieth diagram of one block (see plot.wlai)
plot(ai, subset = list(plot = 'P44005', year = 2001))
```

Index

∗ **data sets**
  Pchron, 29
  Prec, 38
  Prings05, 38
  PTclim05, 45
  Temp, 58
∗ **datasets**
  Pradii03, 37
∗ **package**
  BIOdry-package, 3

amod, 5, *22*, *46*
anova.frametoLme, 6, *60*
anova.lme, *6–8*
anova.modelFrame, 8
arguSelect, 9

BIOdry *(*BIOdry-package*)*, 3
BIOdry-package, 3

cClass, 11, *12*, *16*, *18*, *21*, *46*, *52*, *58*

defineDataRoles, 12, *18*, *21*, *28*, *47*, *51*, *58*, *61*

frametoFriedman, 14, *18*, *22, 23*, *28*, *31*, *33*, *39, 40*, *42*, *54*
frametoLme, *6*, *12, 13*, 15, *21–23*, *28*, *32, 33*, *40*, *43*, *49*, *55*, *57*, *59, 60*
FriedmanForm, *12*, 18, *22, 23*

groupedData, *21*, *26*, *31*

levexp, 19
lme, *8*, *16*, *23*, *60*
lmeForm, *12*, *16*, *18*, 20, *22*, *60*

Map, *46*
mapply, *46*
mgram, *26*

modelFrame, *8*, *18*, *21*, 22, *26*, *28*, *33*, *41*, *44*, *56*, *58*
month.abb, *25*
month.name, *25*
moveYr, 24, *36*, *47*, *49*, *51*, *61*
muleMan, 25, *34*
mUnits, *22*, 27

normalizeGroupLevels, 28

Pchron, 29
plot.default, *36*
plot.frametoFriedman, 30
plot.frametoLme, 32
plot.modelFrame, 33
plot.muleMan, 34
plot.nfnGroupedData, *32, 33*
plot.nmGroupedData, *32, 33*
plot.wlai, 35, *61*
Pradii03, 37
Prec, 38
Prings05, 38
print.frametoFriedman, 39
print.frametoLme, 40
print.modelFrame, 41
print.summary.frametoFriedman, 42
print.summary.frametoLme, 43
print.summary.modelFrame, 44
PTclim05, 45

ringApply, *12*, *18*, *21, 22*, *28*, 46, *49*, *58*
rtimes, *16*, *22, 23*, *46*, 49, *57, 58*

scacum, *22*, *46*, 50
shiftFrame, 51
slitFrame, *46*, 53
summary.frametoFriedman, *39*, 54
summary.frametoLme, *40*, 55
summary.lme, *55*
summary.modelFrame, *41*, 56

supsmu, *14*

tdForm, *12*, *16*, *18*, *22*, *23*, 57, *60*
Temp, 58

update, *23*
update.frametoLme, *7*, 59

wlai, *12*, *24*, *35*, *36*, *46*, 60