Package 'panelr'

October 14, 2022

Title Regression Models and Utilities for Repeated Measures and Panel Data

Version 0.7.6

Description Provides an object type and associated tools for storing and wrangling panel data. Implements several methods for creating regression models that take advantage of the unique aspects of panel data. Among other capabilities, automates the ``within-between'' (also known as ``between-within'' and ``hybrid'') panel regression specification that combines the desirable aspects of both fixed effects and random effects econometric models and fits them as multilevel models (Allison, 2009 <doi:10.4135/9781412993869.d33>; Bell & Jones, 2015 <doi:10.1017/psrm.2014.7>). These models can also be estimated via generalized estimating equations (GEE; McNeish, 2019 <doi:10.1080/00273171.2019.1602504>) and Bayesian estimation is (optionally) supported via 'Stan'. Supports estimation of asymmetric effects models via first differences (Allison, 2019 <doi:10.1177/2378023119826441>) as well as a generalized linear model extension thereof using GEE.

URL https://panelr.jacob-long.com

BugReports https://github.com/jacob-long/panelr

Depends R (>= 3.4.0), lme4

Imports crayon, dplyr, Formula, ggplot2, jtools (>= 2.0.1), lmerTest, magrittr, methods, purrr, rlang (>= 0.3.0), stringr, tibble (>= 2.0.0)

Suggests brms, broom.mixed, car, clubSandwich, geepack, generics, nlme, plm, sandwich, skimr, tidyr (>= 0.8.3), testthat, covr, knitr, rmarkdown

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Jacob A. Long [aut, cre] (<https://orcid.org/0000-0002-1582-6214>)

Maintainer Jacob A. Long <jacob.long@sc.edu>

Repository CRAN

Date/Publication 2021-12-17 07:40:02 UTC

R topics documented:

are_varying *Check if variables are constant or variable over time.*

Description

This function is designed for use with `panel_data()` objects.

Usage

```
are_varying(data, ..., type = "time")
```

Arguments

`data`	A data frame, typically of `panel_data()` class.
`...`	Variable names. If none are given, all variables are checked.
`type`	Check for variance over time or across individuals? Default is `"time"`. `"individual"` considers variables like age to be non-varying because everyone ages at the same speed.

Value

A named logical vector. If TRUE, the variable is varying.

Examples

```
wages <- panel_data(WageData, id = id, wave = t)
wages %>% are_varying(occ, ind, fem, blk)
```

asym *Estimate asymmetric effects models using first differences*

Description

The function fits the asymmetric effects first difference model described in Allison (2019) using GLS estimation.

Usage

```
asym(
  formula,
  data,
  id = NULL,
  wave = NULL,
  use.wave = FALSE,
  min.waves = 1,
  variance = c("toeplitz-1", "constrained", "unconstrained"),
  error.type = c("CR2", "CR1S"),
  ...
)
```

Arguments

formula	Model formula. See details for crucial info on panelr's formula syntax.
data	The data, either a panel_data object or data.frame.
id	If data is not a panel_data object, then the name of the individual id column as a string. Otherwise, leave as NULL, the default.
wave	If data is not a panel_data object, then the name of the panel wave column as a string. Otherwise, leave as NULL, the default.
use.wave	Should the wave be included as a predictor? Default is FALSE.
min.waves	What is the minimum number of waves an individual must have participated in to be included in the analysis? Default is 2 and any valid number is accepted. "all" is also acceptable if you want to include only complete panelists.
variance	One of "toeplitz-1", "constrained", or "unconstrained". The toeplitz variance specification estimates a single error variance and a single lag-1 error correlation with other lags having zero correlation. The constrained model assumes no autocorrelated errors or heteroskedastic errors. The unconstrained option allows separate variances for every period as well as every lag of autocorrelation. This can be very computationally taxing as periods increase and will be inefficient when not necessary. See Allison (2019) for more.
error.type	Either "CR2" or "CR1S". See the clubSandwich package for more details.
...	Ignored.

References

Allison, P. D. (2019). Asymmetric fixed-effects models for panel data. *Socius*, *5*, 1-12. https://doi.org/10.1177/237802311982

Examples

```
## Not run:
data("teen_poverty")
# Convert to long format
teen <- long_panel(teen_poverty, begin = 1, end = 5)
model <- asym(hours ~ lag(pov) + spouse, data = teen)
```

```
summary(model)

## End(Not run)
```

---

asym_gee *Asymmetric effects models fit with GEE*

Description

Fit "within-between" and several other regression variants for panel data via generalized estimating equations.

Usage

```
asym_gee(
  formula,
  data,
  id = NULL,
  wave = NULL,
  cor.str = c("ar1", "exchangeable", "unstructured"),
  use.wave = FALSE,
  wave.factor = FALSE,
  min.waves = 1,
  family = gaussian,
  weights = NULL,
  offset = NULL,
  ...
)
```

Arguments

formula	Model formula. See details for crucial info on panelr's formula syntax.
data	The data, either a panel_data object or data.frame.
id	If data is not a panel_data object, then the name of the individual id column as a string. Otherwise, leave as NULL, the default.
wave	If data is not a panel_data object, then the name of the panel wave column as a string. Otherwise, leave as NULL, the default.
cor.str	Any correlation structure accepted by geepack::geeglm(). Default is "ar1", most useful alternative is "exchangeable". "unstructured" may cause problems due to its computational complexity.
use.wave	Should the wave be included as a predictor? Default is FALSE.
wave.factor	Should the wave variable be treated as an unordered factor instead of continuous? Default is FALSE.

`min.waves`	What is the minimum number of waves an individual must have participated in to be included in the analysis? Default is `2` and any valid number is accepted. `"all"` is also acceptable if you want to include only complete panelists.
`family`	Use this to specify GLM link families. Default is `gaussian`, the linear model.
`weights`	If using weights, either the name of the column in the data that contains the weights or a vector of the weights.
`offset`	this can be used to specify an *a priori* known component to be included in the linear predictor during fitting. This should be `NULL` or a numeric vector of length equal to the number of cases. One or more `offset` terms can be included in the formula instead or as well, and if more than one is specified their sum is used. See `model.offset`.
`...`	Additional arguments provided to `geepack::geeglm()`.

Details

See the documentation for `wbm()` for many details on formula syntax and other arguments.

Value

An `asym_gee` object, which inherits from `wbgee` and `geeglm`.

Author(s)

Jacob A. Long

References

Allison, P. D. (2019). Asymmetric fixed-effects models for panel data. *Socius*, *5*, 1-12. https://doi.org/10.1177/237802311982

McNeish, D. (2019). Effect partitioning in cross-sectionally clustered data without multilevel models. *Multivariate Behavioral Research*, Advance online publication. https://doi.org/10.1080/00273171.2019.1602504

McNeish, D., Stapleton, L. M., & Silverman, R. D. (2016). On the unnecessary ubiquity of hierarchical linear modeling. *Psychological Methods*, *22*, 114-140. https://doi.org/10.1037/met0000078

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
model <- asym_gee(lwage ~ lag(union) + wks, data = wages)
summary(model)
```

`complete_data` *Filter out entities with too few observations*

Description

This function allows you to define a minimum number of waves/periods and exclude all individuals with fewer observations than that.

Usage

```
complete_data(data, ..., formula = NULL, vars = NULL, min.waves = "all")
```

Arguments

`data`	A `panel_data()` frame.
`...`	Optionally, unquoted variable names/expressions separated by commas to be passed to `dplyr::select()`. Otherwise, all columns are included if `formula` and `vars` are also NULL.
`formula`	A formula, like the one you'll be using to specify your model.
`vars`	As an alternative to formula, a vector of variable names.
`min.waves`	What is the minimum number of observations to be kept? Default is `"all"`, but it can be any number.

Details

If `...` (that is, unquoted variable name(s)) are included, then `formula` and `vars` are ignored. Likewise, `formula` takes precedence over `vars`. These are just different methods for selecting variables and you can choose whichever you prefer/are comfortable with. `...` corresponds with the "tidyverse" way, `formula` is useful for programming or working with model formulas, and `vars` is a "standard" evaluation method for when you are working with strings.

Value

A `panel_data` frame.

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
complete_data(wages, wks, lwage, min.waves = 3)
```

`fdm` *Estimate first differences models using GLS*

Description

The function fits first difference models using GLS estimation.

Usage

```
fdm(
  formula,
  data,
  id = NULL,
  wave = NULL,
  use.wave = FALSE,
  min.waves = 1,
  variance = c("toeplitz-1", "constrained", "unconstrained"),
  error.type = c("CR2", "CR1S"),
  ...
)
```

Arguments

`formula`	Model formula. See details for crucial info on `panelr`'s formula syntax.
`data`	The data, either a `panel_data` object or `data.frame`.
`id`	If `data` is not a `panel_data` object, then the name of the individual id column as a string. Otherwise, leave as NULL, the default.
`wave`	If `data` is not a `panel_data` object, then the name of the panel wave column as a string. Otherwise, leave as NULL, the default.
`use.wave`	Should the wave be included as a predictor? Default is FALSE.
`min.waves`	What is the minimum number of waves an individual must have participated in to be included in the analysis? Default is `2` and any valid number is accepted. `"all"` is also acceptable if you want to include only complete panelists.
`variance`	One of `"toeplitz-1"`, `"constrained"`, or `"unconstrained"`. The toeplitz variance specification estimates a single error variance and a single lag-1 error correlation with other lags having zero correlation. The constrained model assumes no autocorrelated errors or heteroskedastic errors. The unconstrained option allows separate variances for every period as well as every lag of autocorrelation. This can be very computationally taxing as periods increase and will be inefficient when not necessary. See Allison (2019) for more.
`error.type`	Either "CR2" or "CR1S". See the `clubSandwich` package for more details.
`...`	Ignored.

References

Allison, P. D. (2019). Asymmetric fixed-effects models for panel data. *Socius*, *5*, 1-12. https://doi.org/10.1177/237802311982

Examples

```
data("teen_poverty")
# Convert to long format
teen <- long_panel(teen_poverty, begin = 1, end = 5)
model <- fdm(hours ~ lag(pov) + spouse, data = teen)
summary(model)
```

`formula.wbm` *Retrieve model formulas from* `wbm` *objects*

Description

This S3 method allows you to retrieve the formula used to fit `wbm` objects.

Usage

```
## S3 method for class 'wbm'
formula(x, raw = FALSE, ...)
```

Arguments

`x`	R object, for `DF2formula()` a `data.frame`.
`raw`	Return the formula used in the call to `lmerMod`/`glmerMod`? Default is FALSE.
`...`	further arguments passed to or from other methods.

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
model <- wbm(lwage ~ lag(union) + wks, data = wages)
# Returns the original model formula rather than the one sent to lme4
formula(model)
```

`get_wave` *Retrieve panel_data metadata*

Description

`get_id()`, `get_wave()`, and `get_periods()` are extractor functions that can be used to retrieve the names of the id and wave variables or time periods of a `panel_data` frame.

Usage

```
get_wave(data)

get_id(data)

get_periods(data)
```

Arguments

data	A panel_data frame

Value

A panel_data frame

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
get_wave(wages)
get_id(wages)
get_periods(wages)
```

---

heise *Estimate Heise stability and reliability coefficients*

Description

This function uses three waves of data to estimate stability and reliability coefficients as described in Heise (1969).

Usage

```
heise(data, ..., waves = NULL)
```

Arguments

data	A panel_data frame.
...	unquoted variable names that are passed to dplyr::select()
waves	Which 3 waves should be used? If NULL (the default), the first, middle, and last waves are used.

Value

A tibble with reliability (rel), waves 1-3 stability (stab13), waves 1-2 stability (stab12), and waves 2-3 stability (stab23) and the variable these values refer to (var).

References

Heise, D. R. (1969). Separating reliability and stability in test-retest correlation. *American Sociological Review*, *34*, 93–101. https://doi.org/10.2307/2092790

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
heise(wages, wks, lwage) # will use waves 1, 4, and 7 by default
```

---

is_panel *Check if object is panel_data*

Description

This is a convenience function that checks whether an object is a `panel_data` object.

Usage

```
is_panel(x)
```

Arguments

`x`	Any object.

Examples

```
data("WageData")
is_panel(WageData) # FALSE
wages <- panel_data(WageData, id = id, wave = t)
is_panel(wages) # TRUE
```

---

line_plot *Plot trends in longitudinal variables*

Description

`line_plot` allows for flexible visualization of repeated measures variables from `panel_data` frames.

Usage

```
line_plot(
  data,
  var,
  id = NULL,
  wave = NULL,
  overlay = TRUE,
  show.points = TRUE,
  subset.ids = FALSE,
  n.random.subset = 9,
  add.mean = FALSE,
  mean.function = "lm",
  line.size = 1,
  alpha = if (overlay) 0.5 else 1
)
```

Arguments

data	Either a panel_data frame or another data frame.
var	The unquoted name of the variable of interest.
id	If data is not a panel_data object, then the id variable.
wave	If data is not a panel_data object, then the wave variable.
overlay	Should the lines be plotted in the same panel or each in their own facet/panel? Default is TRUE, meaning they are plotted in the same panel.
show.points	Plot a point at each wave? Default is TRUE.
subset.ids	Plot only a subset of the entities' lines? Default is NULL, meaning plot all ids. If TRUE, a random subset (the number defined by n.random.subset) are plotted. You may also supply a vector of ids to choose them yourself.
n.random.subset	How many entities to randomly sample when subset.ids is TRUE.
add.mean	Add a line representing the mean trend? Default is FALSE. Cannot be combined with overlay.
mean.function	The mean function to supply to geom_smooth when add.mean is TRUE. Default is "lm", but another option of interest is "loess".
line.size	The thickness of the plotted lines. Default: 0.5
alpha	The transparency for the lines and points. When overlay = TRUE, it is set to 0.5, otherwise 1, which means non-transparent.

Value

The ggplot object.

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
line_plot(wages, lwage, add.mean = TRUE, subset.ids = TRUE, overlay = FALSE)
```

`long_panel` *Convert wide panels to long format*

Description

This function takes wide format panels as input and converts them to long format.

Usage

```
long_panel(
  data,
  prefix = NULL,
  suffix = NULL,
  begin = NULL,
  end = NULL,
  id = "id",
  wave = "wave",
  periods = NULL,
  label_location = c("end", "beginning"),
  as_panel_data = TRUE,
  match = ".*",
  use.regex = FALSE,
  check.varying = TRUE
)
```

Arguments

`data`	The wide data frame.
`prefix`	What character(s) go before the period indicator? If none, set this argument to NULL.
`suffix`	What character(s) go after the period indicator? If none, set this argument to NULL.
`begin`	What is the label for the first period? Could be `1`, `"A"`, or anything that can be sequenced.
`end`	What is the label for the final period? Could be `2`, `"B"`, or anything that can be sequenced and lies further along the sequence than the `begin` argument.
`id`	The name of the ID variable as a string. If there is no ID variable, then this will be the name of the newly-created ID variable.

wave	This will be the name of the newly-created wave variable.
periods	If you period indicator does not lie in a sequence or is not understood by the function, then you can supply them as a vector instead. For instance, you could give `c("one","three","five")` if your variables are labeled `var_one`, `var_three`, and `var_five`.
label_location	Where does the period label go on the variable? If the variables are labeled like `var_1`, `var_2`, etc., then it is `"end"`. If the labels are more like `A_var`, `B_var`, and so on, then it is `"beginning"`.
as_panel_data	Should the return object be a `panel_data()` object? Default is TRUE.
match	The regex that will match the part of the variable names other than the wave indicator. By default it will match any character any amount of times. Sometimes you might know that the variable names should start with a digit, for instance, and you might use `"\\d.*"` instead.
use.regex	Should the `begin` and `end` arguments be treated as regular expressions? Default is FALSE.
check.varying	Should the function check to make sure that every variable in the wide data with a wave indicator is actually time-varying? Default is TRUE, meaning that a constant like "race_W1" only measured in wave 1 will be defined in each wave in the long data. With very large datasets, however, sometimes setting this to FALSE can save memory.

Details

There is no easy way to convert panel data from wide to long format because the both formats are basically non-standard for other applications. This function can handle the common case in which the wide data frame has a regular labeling system for each period. The key thing is providing enough information for the function to understand the pattern.

In the end, this function calls `stats::reshape()` but should be easier to use and able to handle more situations, such as when the label occurs at the beginning of the variable name. Also, just as important, this function has built-in utilities to handle unbalanced data — when variables occur more than once but every single period, which breaks `stats::reshape()`.

Value

Either a `data.frame` or `panel_data` frame.

See Also

`widen_panel()`

Examples

```
## We need a wide data frame, so we will make one from the long-format
## data included in the package.

# Convert WageData to panel_data object
wages <- panel_data(WageData, id = id, wave = t)
```

```
# Convert wages to wide format
wide_wages <- widen_panel(wages)

# Note: wide_wages has variables in the following format:
# var1_1, var1_2, var1_3, var2_1, var2_2, var2_3, etc.
## Not run:
long_wages <- long_panel(wide_wages, prefix = "_", begin = 1, end = 7,
                         id = "id", label_location = "end")

## End(Not run)
# Note that in this case, the prefix and label_location arguments are
# the defaults but are included just for clarity.
```

`make_diff_data` *Generate differenced and asymmetric effects data*

Description

This is an interface to the internal functions that process data for `fdm()`, `asym()`, and `asym_gee()`.

Usage

```
make_diff_data(
  formula,
  data,
  id = NULL,
  wave = NULL,
  use.wave = FALSE,
  min.waves = 1,
  weights = NULL,
  offset = NULL,
  asym = FALSE,
  cumulative = FALSE,
  escape.names = FALSE,
  ...
)
```

Arguments

`formula`	Model formula. See details for crucial info on `panelr`'s formula syntax.
`data`	The data, either a `panel_data` object or `data.frame`.
`id`	If `data` is not a `panel_data` object, then the name of the individual id column as a string. Otherwise, leave as NULL, the default.
`wave`	If `data` is not a `panel_data` object, then the name of the panel wave column as a string. Otherwise, leave as NULL, the default.

use.wave	Should the wave be included as a predictor? Default is FALSE.
min.waves	What is the minimum number of waves an individual must have participated in to be included in the analysis? Default is 2 and any valid number is accepted. "all" is also acceptable if you want to include only complete panelists.
weights	If using weights, either the name of the column in the data that contains the weights or a vector of the weights.
offset	this can be used to specify an *a priori* known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length equal to the number of cases. One or more offset terms can be included in the formula instead or as well, and if more than one is specified their sum is used. See model.offset.
asym	Return asymmetric effects transformed data? Default is FALSE.
cumulative	Return cumulative positive/negative differences, most useful for fixed effects estimation and/or generalized linear models? Default is FALSE.
escape.names	Return only syntactically valid variable names? Default is FALSE.
...	Ignored.

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
make_diff_data(wks ~ lwage + union, data = wages)
```

---

make_wb_data *Prepare data for within-between modeling*

Description

This function allows users to make the changes to their data that occur in wbm() without having to fit the model.

Usage

```
make_wb_data(
  formula,
  data,
  id = NULL,
  wave = NULL,
  model = "w-b",
  detrend = FALSE,
  use.wave = FALSE,
  wave.factor = FALSE,
  min.waves = 2,
```

```
  balance.correction = FALSE,
  dt.random = TRUE,
  dt.order = 1,
  weights = NULL,
  offset = NULL,
  interaction.style = c("double-demean", "demean", "raw"),
  ...
)
```

Arguments

formula	Model formula. See details for crucial info on panelr's formula syntax.
data	The data, either a panel_data object or data.frame.
id	If data is not a panel_data object, then the name of the individual id column as a string. Otherwise, leave as NULL, the default.
wave	If data is not a panel_data object, then the name of the panel wave column as a string. Otherwise, leave as NULL, the default.
model	One of "w-b", "within", "between", "contextual". See details for more on these options.
detrend	Adjust within-subject effects for trends in the predictors? Default is FALSE, but some research suggests this is a better idea (see Curran and Bauer (2011) reference).
use.wave	Should the wave be included as a predictor? Default is FALSE.
wave.factor	Should the wave variable be treated as an unordered factor instead of continuous? Default is FALSE.
min.waves	What is the minimum number of waves an individual must have participated in to be included in the analysis? Default is 2 and any valid number is accepted. "all" is also acceptable if you want to include only complete panelists.
balance.correction	Correct between-subject effects for unbalanced panels following the procedure in Curran and Bauer (2011)? Default is FALSE.
dt.random	Should the detrending procedure be performed with a random slope for each entity? Default is TRUE but for short panels FALSE may be better, fitting a trend for all entities.
dt.order	If detrending using detrend, what order polynomial would you like to specify for the relationship between time and the predictors? Default is 1, a linear model.
weights	If using weights, either the name of the column in the data that contains the weights or a vector of the weights.
offset	this can be used to specify an *a priori* known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length equal to the number of cases. One or more offset terms can be included in the formula instead or as well, and if more than one is specified their sum is used. See model.offset.

interaction.style
: The best way to calculate interactions in within models is in some dispute. The conventional way ("demean") is to first calculate the product of the variables involved in the interaction before those variables have their means subtracted and then subtract the mean of the product from the product term (see Schunk and Perales (2017)). Giesselmann and Schmidt-Catran (2020) show this method carries between-entity differences that within models are designed to model out. They suggest an alternate method ("double-demean") in which the product term is first calculated using the de-meaned lower-order variables and then the subject means are subtracted from this product term. Another option is to simply use the product term of the de-meaned variables ("raw"), but Giesselmann and Schmidt-Catran (2020) show this method biases the results towards zero effect. The default is "double-demean" but if emulating other software is the goal, "demean" might be preferred.

...
: Additional arguments provided to lme4::lmer(), lme4::glmer(), or lme4::glmer.nb().

Value

A panel_data object with the requested specification.

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
make_wb_data(lwage ~ wks + union | fem, data = wages)
```

---

model_frame *Make model frames for panel_data objects*

Description

This is similar to model.frame, but is designed specifically for panel_data() data frames. It's a workhorse in wbm() but may be useful in scripting use as well.

Usage

```
model_frame(formula, data)
```

Arguments

formula
: A formula. Note that to get an individual-level mean with incomplete data (e.g., panel attrition), you should use imean() rather than mean().

data
: A panel_data() frame.

Value

A panel_data() frame with only the columns needed to fit a model as described by the formula.

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
model_frame(lwage ~ wks + exp, data = wages)
```

`nlsy` *National Longitudinal Survey of Youth data*

Description

These data come from the years 1990-1994 in the National Longitudinal Survey of Youth, with information about 581 individuals. These data are in the "wide" format for demonstration purposes.

Usage

```
nlsy
```

Format

A data frame with 581 rows and 16 variables:

momage Mother's age at birth

gender 0 if boy, 1 if girl

momwork 1 if mother works, 0 if not)

married 1 if parents are married, 0 if not

hispanic 1 if child is Hispanic, 0 if not

black 1 if child is black, 0 if not

childage Child's age at first interview

anti90 A measure of anti-social behavior antisocial behavior measured on a scale from 0 to 6, taken in 1990

anti92 A measure of anti-social behavior antisocial behavior measured on a scale from 0 to 6, taken in 1992

anti94 A measure of anti-social behavior antisocial behavior measured on a scale from 0 to 6, taken in 1994

self90 A measure of self-esteem measured on a scale from 6 to 24, taken in 1990

self92 A measure of self-esteem measured on a scale from 6 to 24, taken in 1992

self94 A measure of self-esteem measured on a scale from 6 to 24, taken in 1994

pov90 1 if family is in poverty, 0 if not, in 1990

pov92 1 if family is in poverty, 0 if not, in 1992

pov94 1 if family is in poverty, 0 if not, in 1994

Source

These data originate with the U.S. Department of Labor. The particular subset used here come from Paul Allison via Statistical Horizons: https://statisticalhorizons.com/wp-content/uploads/nlsy.dta

---

nobs.wbm *Number of observations used in* wbm *models*

Description

This S3 method allows you to retrieve either the number of observations or number of entities in the data used to fit wbm objects.

Usage

```
## S3 method for class 'wbm'
nobs(object, entities = TRUE, ...)
```

Arguments

object	A fitted model object.
entities	Should nobs return the number of entities in the panel or the number of rows in the panel_data frame? Default is TRUE, returning the number of entities.
...	Further arguments to be passed to methods.

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
model <- wbm(lwage ~ lag(union) + wks, data = wages)
nobs(model)
```

---

panel_data *Create panel data frames*

Description

Format your data for use with **panelr**.

Usage

```
panel_data(data, id = id, wave = wave, ...)

as_pdata.frame(data)

as_panel_data(data, ...)

## Default S3 method:
as_panel_data(data, id = id, wave = wave, ...)

## S3 method for class 'pdata.frame'
as_panel_data(data, ...)

as_panel(data, ...)
```

Arguments

data	A data frame.
id	The name of the column (unquoted) that identifies participants/entities. A new column will be created called id, overwriting any column that already has that name.
wave	The name of the column (unquoted) that identifies waves or periods. A new column will be created called wave, overwriting any column that already has that name.
...	Attributes for adding onto this method. See tibble::new_tibble() for a run-through of the logic.

Value

A panel_data object.

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
```

---

predict.wbgee *Predictions and simulations from within-between GEE models*

Description

These methods facilitate fairly straightforward predictions from wbgee models.

Usage

```
## S3 method for class 'wbgee'
predict(
  object,
  newdata = NULL,
  se.fit = FALSE,
  raw = FALSE,
  type = c("link", "response"),
  ...
)
```

Arguments

object	Object of class inheriting from "lm"
newdata	An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
se.fit	A switch indicating if standard errors are required.
raw	Is newdata a geeglm model frame or panel_data? TRUE indicates a geeglm-style newdata, with all of the extra columns created by wbgee.
type	Type of prediction (response or model term). Can be abbreviated.
...	further arguments passed to or from other methods.

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
model <- wbgee(lwage ~ lag(union) + wks, data = wages)
# By default, assumes you're using the processed data for newdata
predict(model)
```

---

predict.wbm *Predictions and simulations from within-between models*

Description

These methods facilitate fairly straightforward predictions and simulations from wbm models.

Usage

```
## S3 method for class 'wbm'
predict(
  object,
  newdata = NULL,
  se.fit = FALSE,
  raw = FALSE,
  use.re.var = FALSE,
```

```
  re.form = NULL,
  type = c("link", "response"),
  allow.new.levels = TRUE,
  na.action = na.pass,
  ...
)

## S3 method for class 'wbm'
simulate(
  object,
  nsim = 1,
  seed = NULL,
  use.u = FALSE,
  newdata = NULL,
  raw = FALSE,
  newparams = NULL,
  re.form = NA,
  type = c("link", "response"),
  allow.new.levels = FALSE,
  na.action = na.pass,
  ...
)
```

Arguments

object	a fitted model object
newdata	data frame for which to evaluate predictions.
se.fit	Include standard errors with the predictions? Note that these standard errors by default include only fixed effects variance. See details for more info. Default is FALSE.
raw	Is newdata a merMod model frame or panel_data? TRUE indicates a merMod-style newdata, with all of the extra columns created by wbm.
use.re.var	If se.fit is TRUE, include random effects variance in standard errors? Default is FALSE.
re.form	(formula, NULL, or NA) specify which random effects to condition on when predicting. If NULL, include all random effects; if NA or ~0, include no random effects.
type	character string - either "link", the default, or "response" indicating the type of prediction object returned.
allow.new.levels	logical if new levels (or NA values) in newdata are allowed. If FALSE (default), such new values in newdata will trigger an error; if TRUE, then the prediction will use the unconditional (population-level) values for data with previously unobserved levels (or NAs).
na.action	function determining what should be done with missing values for fixed effects in newdata. The default is to predict NA: see na.pass.

...	When boot and se.fit are TRUE, any additional arguments are passed to lme4::bootMer().
nsim	positive integer scalar - the number of responses to simulate.
seed	an optional seed to be used in set.seed immediately before the simulation so as to generate a reproducible sample.
use.u	(logical) if TRUE, generate a simulation conditional on the current random-effects estimates; if FALSE generate new Normally distributed random-effects values. (Redundant with re.form, which is preferred: TRUE corresponds to re.form = NULL (condition on all random effects), while FALSE corresponds to re.form = ~0 (condition on none of the random effects).)
newparams	new parameters to use in evaluating predictions, specified as in the start parameter for lmer or glmer – a list with components theta and beta and (for LMMs or GLMMs that estimate a scale parameter) sigma

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
model <- wbm(lwage ~ lag(union) + wks, data = wages)
# By default, assumes you're using the processed data for newdata
predict(model)
```

---

summary.panel_data *Summarize panel data frames*

Description

summary method for panel_data objects.

Usage

```
## S3 method for class 'panel_data'
summary(object, ..., by.wave = TRUE, by.id = FALSE)
```

Arguments

object	A panel_data frame.
...	Optionally, unquoted variable names/expressions separated by commas to be passed to dplyr::select(). Otherwise, all columns are included.
by.wave	(if skimr is installed) Separate descriptives by wave? Default is TRUE.
by.id	(if skimr is installed) Separate descriptives by entity? Default is FALSE. Be careful if you have a large number of entities as the output will be massive.

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
summary(wages, lwage, exp, wks)
```

`teen_poverty` *National Longitudinal Survey of Youth teenage women poverty data*

Description

These data come from the years 1979-1983 in the National Longitudinal Survey of Youth, with information about 1141 teenage women. These data are in the "wide" format for demonstration purposes.

Usage

```
teen_poverty
```

Format

A data frame with 1141 rows and 28 variables:

id Unique identifier for the respondent

age Age at first interview

black 1 if subject is black, 0 if not

pov1 1 if subject is in poverty, 0 if not, at time 1

pov2 1 if subject is in poverty, 0 if not, at time 2

pov3 1 if subject is in poverty, 0 if not, at time 3

pov4 1 if subject is in poverty, 0 if not, at time 4

pov5 1 if subject is in poverty, 0 if not, at time 5

mother1 1 if subject has had a child, 0 if not, at time 1

mother2 1 if subject has had a child, 0 if not, at time 2

mother3 1 if subject has had a child, 0 if not, at time 3

mother4 1 if subject has had a child, 0 if not, at time 4

mother5 1 if subject has had a child, 0 if not, at time 5

spouse1 1 if subject lives with a spouse, 0 if not, at time 1

spouse2 1 if subject lives with a spouse, 0 if not, at time 2

spouse3 1 if subject lives with a spouse, 0 if not, at time 3

spouse4 1 if subject lives with a spouse, 0 if not, at time 4

spouse5 1 if subject lives with a spouse, 0 if not, at time 5

inschool1 1 if subject is in school, 0 if not, at time 1

inschool2 1 if subject is in school, 0 if not, at time 2

inschool3 1 if subject is in school, 0 if not, at time 3

inschool4 1 if subject is in school, 0 if not, at time 4

inschool5 1 if subject is in school, 0 if not, at time 5

hours1 Hours worked during the week of the survey, at time 1

hours2 Hours worked during the week of the survey, at time 2

hours3 Hours worked during the week of the survey, at time 3

hours4 Hours worked during the week of the survey, at time 4

hours5 Hours worked during the week of the survey, at time 5

Source

These data originate with the U.S. Department of Labor. The particular subset used here come from Paul Allison via Statistical Horizons: https://statisticalhorizons.com/wp-content/uploads/teenpov.dta

---

`tidy.asym` *Tidy methods for* `fdm` *and* `asym` *models*

Description

`panelr` provides methods to access `fdm` and `asym` data in a tidy format

Usage

```
## S3 method for class 'asym'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)

## S3 method for class 'fdm'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)

## S3 method for class 'fdm'
glance(x, ...)
```

Arguments

`x`	An `fdm` or `asym` object.
`conf.int`	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to `FALSE`.
`conf.level`	The confidence level to use for the confidence interval if `conf.int = TRUE`. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
`...`	Ignored

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
model <- fdm(lwage ~ wks + union, data = wages)
if (requireNamespace("generics")) {
  generics::tidy(model)
}
```

`tidy.asym_gee` *Tidy methods for* `wbgee` *models*

Description

`panelr` provides methods to access `wbgee` data in a tidy format

Usage

```
## S3 method for class 'asym_gee'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)

## S3 method for class 'wbgee'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)

## S3 method for class 'wbgee'
glance(x, ...)
```

Arguments

`x`	A `wbgee` object.
`conf.int`	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to `FALSE`.
`conf.level`	The confidence level to use for the confidence interval if `conf.int = TRUE`. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
`...`	Ignored

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
model <- wbgee(lwage ~ lag(union) + wks, data = wages)
if (requireNamespace("generics")) {
  generics::tidy(model)
}
```

tidy.wbm *Tidy methods for* wbm *models*

Description

panelr provides methods to access wbm data in a tidy format

Usage

```
## S3 method for class 'wbm'
tidy(
  x,
  conf.int = FALSE,
  conf.level = 0.95,
  effects = c("fixed", "ran_pars"),
  conf.method = "Wald",
  ran_prefix = NULL,
  ...
)

## S3 method for class 'wbm'
glance(x, ...)

## S3 method for class 'summ.wbm'
glance(x, ...)

## S3 method for class 'summ.wbm'
tidy(x, ...)
```

Arguments

x	An object of class merMod, such as those from lmer, glmer, or nlmer
conf.int	whether to include a confidence interval
conf.level	confidence level for CI
effects	A character vector including one or more of "fixed" (fixed-effect parameters); "ran_pars" (variances and covariances or standard deviations and correlations of random effect terms); "ran_vals" (conditional modes/BLUPs/latent variable estimates); or "ran_coefs" (predicted parameter values for each group, as returned by coef.merMod.
conf.method	method for computing confidence intervals (see lme4::confint.merMod)
ran_prefix	a length-2 character vector specifying the strings to use as prefixes for self- (variance/standard deviation) and cross- (covariance/correlation) random effects terms
...	Additional arguments (passed to confint.merMod for tidy; augment_columns for augment; ignored for glance)

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
model <- wbm(lwage ~ lag(union) + wks, data = wages)
if (requireNamespace("broom.mixed")) {
  broom.mixed::tidy(model)
}
```

---

unpanel *Convert panel_data to regular data frame*

Description

This convenience function removes the special features of `panel_data`.

Usage

```
unpanel(panel)
```

Arguments

`panel`	A `panel_data` object.

Value

An ungrouped `tibble`.

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
wages_non_panel <- unpanel(wages)
```

WageData *Earnings data from the Panel Study of Income Dynamics*

Description

These data come from the years 1976-1982 in the Panel Study of Income Dynamics (PSID), with information about the demographics and earnings of 595 individuals.

Usage

```
WageData
```

Format

A data frame with 4165 rows and 14 variables:

id Unique identifier for each survey respondent

t A number corresponding to each wave of the survey, 1 through 7

wks Weeks worked in the past year

lwage Natural logarithm of earnings in the past year

union Binary indicator whether respondent is a member of union (1 = union member)

ms Binary indicator for whether respondent is married (1 = married)

occ Binary indicator for whether respondent is a blue collar (= 0) or white collar (= 1) worker.

ind Binary indicator for whether respondent works in manufacturing (= 1)

south Binary indicator for whether respondent lives in the South (= 1)

smsa Binary indicator for whether respondent lives in a standard metropolitan area (SMSA; = 1)

fem Binary indicator for whether respondent is female (= 1)

blk Binary indicator for whether respondent is African-American (= 1)

ed Years of education

exp Years in the workforce.

Source

These data are all over the place. This particular file was downloaded from Richard Williams at http://www3.nd.edu/~rwilliam/statafiles/wages.dta, though he doesn't claim ownership of these data.

The data were shared as a supplement to Baltagi (2005) at https://www.wiley.com/legacy/wileychi/baltagi3e/data_sets.html.

They were also shared as a supplement to Greene (2008) at http://pages.stern.nyu.edu/~wgreene/Text/Edition6/tablelist6.htm.

The data are also available in numerous other locations, including in slightly different formats as Wages in the **plm** package and PSID7682 in the **AER** package.

---

wbgee *Panel regression models fit with GEE*

Description

Fit "within-between" and several other regression variants for panel data via generalized estimating equations.

Usage

```
wbgee(
  formula,
  data,
  id = NULL,
  wave = NULL,
  model = "w-b",
  cor.str = c("ar1", "exchangeable", "unstructured"),
  detrend = FALSE,
  use.wave = FALSE,
  wave.factor = FALSE,
  min.waves = 2,
  family = gaussian,
  balance.correction = FALSE,
  dt.random = TRUE,
  dt.order = 1,
  weights = NULL,
  offset = NULL,
  interaction.style = c("double-demean", "demean", "raw"),
  scale = FALSE,
  scale.response = FALSE,
  n.sd = 1,
  calc.fit.stats = TRUE,
  ...
)
```

Arguments

formula	Model formula. See details for crucial info on panelr's formula syntax.
data	The data, either a panel_data object or data.frame.
id	If data is not a panel_data object, then the name of the individual id column as a string. Otherwise, leave as NULL, the default.
wave	If data is not a panel_data object, then the name of the panel wave column as a string. Otherwise, leave as NULL, the default.
model	One of "w-b", "within", "between", "contextual". See details for more on these options.
cor.str	Any correlation structure accepted by geepack::geeglm(). Default is "ar1", most useful alternative is "exchangeable". "unstructured" may cause problems due to its computational complexity.
detrend	Adjust within-subject effects for trends in the predictors? Default is FALSE, but some research suggests this is a better idea (see Curran and Bauer (2011) reference).
use.wave	Should the wave be included as a predictor? Default is FALSE.
wave.factor	Should the wave variable be treated as an unordered factor instead of continuous? Default is FALSE.

`min.waves` What is the minimum number of waves an individual must have participated in to be included in the analysis? Default is `2` and any valid number is accepted. `"all"` is also acceptable if you want to include only complete panelists.

`family` Use this to specify GLM link families. Default is `gaussian`, the linear model.

`balance.correction`
Correct between-subject effects for unbalanced panels following the procedure in Curran and Bauer (2011)? Default is FALSE.

`dt.random` Should the detrending procedure be performed with a random slope for each entity? Default is TRUE but for short panels FALSE may be better, fitting a trend for all entities.

`dt.order` If detrending using `detrend`, what order polynomial would you like to specify for the relationship between time and the predictors? Default is 1, a linear model.

`weights` If using weights, either the name of the column in the data that contains the weights or a vector of the weights.

`offset` this can be used to specify an *a priori* known component to be included in the linear predictor during fitting. This should be `NULL` or a numeric vector of length equal to the number of cases. One or more `offset` terms can be included in the formula instead or as well, and if more than one is specified their sum is used. See `model.offset`.

`interaction.style`
The best way to calculate interactions in within models is in some dispute. The conventional way (`"demean"`) is to first calculate the product of the variables involved in the interaction before those variables have their means subtracted and then subtract the mean of the product from the product term (see Schunk and Perales (2017)). Giesselmann and Schmidt-Catran (2020) show this method carries between-entity differences that within models are designed to model out. They suggest an alternate method (`"double-demean"`) in which the product term is first calculated using the de-meaned lower-order variables and then the subject means are subtracted from this product term. Another option is to simply use the product term of the de-meaned variables (`"raw"`), but Giesselmann and Schmidt-Catran (2020) show this method biases the results towards zero effect. The default is `"double-demean"` but if emulating other software is the goal, `"demean"` might be preferred.

`scale` If `TRUE`, reports standardized regression coefficients. Default is `FALSE`.

`scale.response` Should the response variable also be rescaled? Default is `FALSE`.

`n.sd` How many standard deviations should you divide by for standardization? Default is 1, though some prefer 2.

`calc.fit.stats` Calculate fit statistics? Default is TRUE, but occasionally poor-fitting models might trip up here.

`...` Additional arguments provided to `geepack::geeglm()`.

Details

See the documentation for `wbm()` for many details on formula syntax and other arguments.

Value

A wbgee object, which inherits from geeglm.

Author(s)

Jacob A. Long

References

Allison, P. (2009). *Fixed effects regression models*. Thousand Oaks, CA: SAGE Publications. https://doi.org/10.4135/9781412993869.d33

Bell, A., & Jones, K. (2015). Explaining fixed effects: Random effects modeling of time-series cross-sectional and panel data. *Political Science Research and Methods*, *3*, 133–153. https://doi.org/10.1017/psrm.2014.7

Curran, P. J., & Bauer, D. J. (2011). The disaggregation of within-person and between-person effects in longitudinal models of change. *Annual Review of Psychology*, *62*, 583–619. https://doi.org/10.1146/annurev.psych.09

Giesselmann, M., & Schmidt-Catran, A. W. (2020). Interactions in fixed effects regression models. *Sociological Methods & Research*, 1–28. https://doi.org/10.1177/0049124120914934

McNeish, D. (2019). Effect partitioning in cross-sectionally clustered data without multilevel models. *Multivariate Behavioral Research*, Advance online publication. https://doi.org/10.1080/00273171.2019.1602504

McNeish, D., Stapleton, L. M., & Silverman, R. D. (2016). On the unnecessary ubiquity of hierarchical linear modeling. *Psychological Methods*, *22*, 114-140. https://doi.org/10.1037/met0000078

Schunck, R., & Perales, F. (2017). Within- and between-cluster effects in generalized linear mixed models: A discussion of approaches and the xthybrid command. *The Stata Journal*, *17*, 89–115. https://doi.org/10.1177/1536867X1701700106

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
model <- wbgee(lwage ~ lag(union) + wks | blk + fem | blk * lag(union),
         data = wages)
summary(model)
```

---

wbm *Panel regression models fit via multilevel modeling*

Description

Fit "within-between" and several other regression variants for panel data in a multilevel modeling framework.

Usage

```
wbm(
  formula,
  data,
  id = NULL,
  wave = NULL,
  model = "w-b",
  detrend = FALSE,
  use.wave = FALSE,
  wave.factor = FALSE,
  min.waves = 2,
  family = gaussian,
  balance.correction = FALSE,
  dt.random = TRUE,
  dt.order = 1,
  pR2 = TRUE,
  pvals = TRUE,
  t.df = "Satterthwaite",
  weights = NULL,
  offset = NULL,
  interaction.style = c("double-demean", "demean", "raw"),
  scale = FALSE,
  scale.response = FALSE,
  n.sd = 1,
  dt_random = dt.random,
  dt_order = dt.order,
  balance_correction = balance.correction,
  ...
)
```

Arguments

formula	Model formula. See details for crucial info on panelr's formula syntax.
data	The data, either a panel_data object or data.frame.
id	If data is not a panel_data object, then the name of the individual id column as a string. Otherwise, leave as NULL, the default.
wave	If data is not a panel_data object, then the name of the panel wave column as a string. Otherwise, leave as NULL, the default.
model	One of "w-b", "within", "between", "contextual". See details for more on these options.
detrend	Adjust within-subject effects for trends in the predictors? Default is FALSE, but some research suggests this is a better idea (see Curran and Bauer (2011) reference).
use.wave	Should the wave be included as a predictor? Default is FALSE.
wave.factor	Should the wave variable be treated as an unordered factor instead of continuous? Default is FALSE.

`min.waves` What is the minimum number of waves an individual must have participated in to be included in the analysis? Default is `2` and any valid number is accepted. `"all"` is also acceptable if you want to include only complete panelists.

`family` Use this to specify GLM link families. Default is `gaussian`, the linear model.

`balance.correction`
Correct between-subject effects for unbalanced panels following the procedure in Curran and Bauer (2011)? Default is FALSE.

`dt.random` Should the detrending procedure be performed with a random slope for each entity? Default is TRUE but for short panels FALSE may be better, fitting a trend for all entities.

`dt.order` If detrending using `detrend`, what order polynomial would you like to specify for the relationship between time and the predictors? Default is 1, a linear model.

`pR2` Calculate a pseudo R-squared? Default is TRUE, but in some cases may cause errors or add computation time.

`pvals` Calculate p values? Default is TRUE but for some complex linear models, this may take a long time to compute using the `pbkrtest` package.

`t.df` For linear models only. User may choose the method for calculating the degrees of freedom in t-tests. Default is `"Satterthwaite"`, but you may also choose `"Kenward-Roger"`. Kenward-Roger standard errors/degrees of freedom requires the `pbkrtest` package.

`weights` If using weights, either the name of the column in the data that contains the weights or a vector of the weights.

`offset` this can be used to specify an *a priori* known component to be included in the linear predictor during fitting. This should be `NULL` or a numeric vector of length equal to the number of cases. One or more `offset` terms can be included in the formula instead or as well, and if more than one is specified their sum is used. See `model.offset`.

`interaction.style`
The best way to calculate interactions in within models is in some dispute. The conventional way (`"demean"`) is to first calculate the product of the variables involved in the interaction before those variables have their means subtracted and then subtract the mean of the product from the product term (see Schunk and Perales (2017)). Giesselmann and Schmidt-Catran (2020) show this method carries between-entity differences that within models are designed to model out. They suggest an alternate method (`"double-demean"`) in which the product term is first calculated using the de-meaned lower-order variables and then the subject means are subtracted from this product term. Another option is to simply use the product term of the de-meaned variables (`"raw"`), but Giesselmann and Schmidt-Catran (2020) show this method biases the results towards zero effect. The default is `"double-demean"` but if emulating other software is the goal, `"demean"` might be preferred.

`scale` If `TRUE`, reports standardized regression coefficients. Default is `FALSE`.

`scale.response` Should the response variable also be rescaled? Default is `FALSE`.

`n.sd` How many standard deviations should you divide by for standardization? Default is 1, though some prefer 2.

dt_random	Deprecated. Equivalent to dt.random.
dt_order	Deprecated. Equivalent to dt.order.
balance_correction	Deprecated. Equivalent to balance.correction.
...	Additional arguments provided to lme4::lmer(), lme4::glmer(), or lme4::glmer.nb().

Details

Formula syntax

The within-between models, and multilevel panel models more generally, distinguish between time-varying and time-invariant predictors. These are, as they sound, variables that are either measured repeatedly (in every wave) in the case of time-varying predictors or only once in the case of time-invariant predictors. You need to specify these separately in the formula to tell the model which variables you expect to change over time and which will not. The primary way of doing so is via the | operator.

As an example, we can look at the WageData included in this package. We will create a model that predicts the logarithm of the individual's wages (lwage) with their union status (union), which can change over time, and their race (blk; dichotomized as black or non-black), which does not change throughout the period of study. Our formula will look like this:

```
lwage ~ union | blk
```

Put time-varying variables before the first | and time-invariant variables afterwards. You can specify lags like lag(union) for time-varying variables; for more than 1 lag, include the number: lag(union, 2).

After the first | go the time-invariant variables. Note that if you put a time-varying variable here, what you get is the observed value rather than one adjusted to isolate within-entity effects. You may also take a time-varying variable — let's say weeks worked (wks) — and use imean(wks) to include the individual's mean across all waves as a predictor while omitting the per-wave measures.

There is also a place for a second |. Here you can specify cross-level interactions (within-level interactions can be specified here as well). If I wanted the interaction term for union and blk — to see whether the effect of union status depended on one's race — I would specify the formula this way:

```
lwage ~ union | blk | union * blk
```

Another use for the post-second | section of the formula is for changing the random effects specification. By default, only a random intercept is specified in the call to lme4::lmer()/lme4::glmer(). If you would like to specify other random slopes, include them here using the typical lme4 syntax:

```
lwage ~ union | blk | (union | id)
```

You can also include the wave variable in a random effects term to specify a latent growth curve model:

```
lwage ~ union | blk + t | (t | id)
```

One last thing to know: If you want to use the second | but not the first, put a 1 or 0 after the first, like this:

```
lwage ~ union | 1 | (union | id)
```

Of course, with no time-invariant variables, you need no | operators at all.

Models

As a convenience, wbm does the heavy lifting for specifying the within-between model correctly. As a side effect it only takes a few easy tweaks to specify the model slightly differently. You can change this behavior with the model argument.

By default, the argument is "w-b" (equivalently, "within-between"). This means, for each time-varying predictor, you have two types of variables in the model. The "between" effect is represented by the individual-level mean for each entity (e.g., each respondent to a panel survey). The "within" effect is represented by each wave's measure *with the individual-level mean* subtracted. Some refer to this as "de-meaning." Thinking in a Hausman test framework — with the within-between model as described here — you should expect the within and between coefficients to be the same if a random effects model were appropriate.

The contextual model is very similar (use argument "contextual"). In some situations, this will be more intuitive to interpret. Empirically, the only difference compared to the within-between specification is that the contextual model does not subtract the individual-level means from the wave-level measures. This also changes the interpretation of the between-subject coefficients: In the contextual model, they are the *difference* between the within and between effects. If there's no difference between within and between effects, then, the coefficients will be 0.

To fit a random effects model, use either "between" or "random". This involves no de-meaning and no individual-level means whatsoever.

To fit a fixed effects model, use either "within" or "fixed". Any between-subjects terms in the formula will be ignored. The time-varying variables will be de-meaned, but the individual-level mean is not included in the model.

Value

A wbm object, which inherits from merMod.

Author(s)

Jacob A. Long

References

Allison, P. (2009). *Fixed effects regression models*. Thousand Oaks, CA: SAGE Publications. https://doi.org/10.4135/9781412993869.d33

Bell, A., & Jones, K. (2015). Explaining fixed effects: Random effects modeling of time-series cross-sectional and panel data. *Political Science Research and Methods*, *3*, 133–153. https://doi.org/10.1017/psrm.2014.7

Curran, P. J., & Bauer, D. J. (2011). The disaggregation of within-person and between-person effects in longitudinal models of change. *Annual Review of Psychology*, *62*, 583–619. https://doi.org/10.1146/annurev.psych.09

Giesselmann, M., & Schmidt-Catran, A. (2018). Interactions in fixed effects regression models (Discussion Papers of DIW Berlin No. 1748). *DIW Berlin, German Institute for Economic Research*. Retrieved from https://ideas.repec.org/p/diw/diwwpp/dp1748.html

Schunck, R., & Perales, F. (2017). Within- and between-cluster effects in generalized linear mixed models: A discussion of approaches and the xthybrid command. *The Stata Journal*, *17*, 89–115. https://doi.org/10.1177/1536867X1701700106

See Also

wbm_stan() for a Bayesian estimation option.

Examples

```
data("WageData")
wages <- panel_data(WageData, id = id, wave = t)
model <- wbm(lwage ~ lag(union) + wks | blk + fem | blk * lag(union),
         data = wages)
summary(model)
```

---

wbm-class *Within-Between Model (wbm) class*

Description

Models fit using wbm() return values of this class, which inherits from merMod-class.

Slots

call_info A list of metadata about the arguments used.

call The actual function call.

summ The jtools::summ() object returned from calling it on the merMod object.

summ_atts The attributes of the summ object.

orig_data The data provided to the data argument in the function call.

---

wbm_stan *Bayesian estimation of within-between models*

Description

A near-equivalent of wbm() that instead uses Stan, via **rstan** and **brms**.

Usage

```
wbm_stan(
  formula,
  data,
  id = NULL,
  wave = NULL,
  model = "w-b",
  detrend = FALSE,
  use.wave = FALSE,
```

```
  wave.factor = FALSE,
  min.waves = 2,
  model.cor = FALSE,
  family = gaussian,
  fit_model = TRUE,
  balance.correction = FALSE,
  dt.random = TRUE,
  dt.order = 1,
  chains = 3,
  iter = 2000,
  scale = FALSE,
  save_ranef = FALSE,
  interaction.style = c("double-demean", "demean", "raw"),
  weights = NULL,
  offset = NULL,
  ...
)
```

Arguments

formula	Model formula. See details for crucial info on panelr's formula syntax.
data	The data, either a panel_data object or data.frame.
id	If data is not a panel_data object, then the name of the individual id column as a string. Otherwise, leave as NULL, the default.
wave	If data is not a panel_data object, then the name of the panel wave column as a string. Otherwise, leave as NULL, the default.
model	One of "w-b", "within", "between", "contextual". See details for more on these options.
detrend	Adjust within-subject effects for trends in the predictors? Default is FALSE, but some research suggests this is a better idea (see Curran and Bauer (2011) reference).
use.wave	Should the wave be included as a predictor? Default is FALSE.
wave.factor	Should the wave variable be treated as an unordered factor instead of continuous? Default is FALSE.
min.waves	What is the minimum number of waves an individual must have participated in to be included in the analysis? Default is 2 and any valid number is accepted. "all" is also acceptable if you want to include only complete panelists.
model.cor	Do you want to model residual autocorrelation? This is often appropriate for linear models (family = gaussian). Default is FALSE to be consistent with wbm(), reduce runtime, and avoid warnings for non-linear models.
family	Use this to specify GLM link families. Default is gaussian, the linear model.
fit_model	Fit the model? Default is TRUE. If FALSE, only the model code is returned.
balance.correction	Correct between-subject effects for unbalanced panels following the procedure in Curran and Bauer (2011)? Default is FALSE.

dt.random — Should the detrending procedure be performed with a random slope for each entity? Default is TRUE but for short panels FALSE may be better, fitting a trend for all entities.

dt.order — If detrending using detrend, what order polynomial would you like to specify for the relationship between time and the predictors? Default is 1, a linear model.

chains — How many Markov chains should be used? Default is 3, to leave you with one unused thread if you're on a typical dual-core machine.

iter — How many iterations, including warmup? Default is 2000, leaving 1000 per chain after warmup. For some models and data, you may need quite a few more.

scale — Standardize predictors? This can speed up model fit. Default is FALSE.

save_ranef — Save random effect estimates? This can be crucial for predicting from the model and for certain post-estimation procedures. On the other hand, it drastically increases the size of the resulting model. Default is FALSE.

interaction.style — The best way to calculate interactions in within models is in some dispute. The conventional way ("demean") is to first calculate the product of the variables involved in the interaction before those variables have their means subtracted and then subtract the mean of the product from the product term (see Schunk and Perales (2017)). Giesselmann and Schmidt-Catran (2020) show this method carries between-entity differences that within models are designed to model out. They suggest an alternate method ("double-demean") in which the product term is first calculated using the de-meaned lower-order variables and then the subject means are subtracted from this product term. Another option is to simply use the product term of the de-meaned variables ("raw"), but Giesselmann and Schmidt-Catran (2020) show this method biases the results towards zero effect. The default is "double-demean" but if emulating other software is the goal, "demean" might be preferred.

weights — If using weights, either the name of the column in the data that contains the weights or a vector of the weights.

offset — this can be used to specify an *a priori* known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length equal to the number of cases. One or more offset terms can be included in the formula instead or as well, and if more than one is specified their sum is used. See model.offset.

... — Additional arguments passed on to brms::brm(). This can include specification of priors.

Details

See wbm() for details on the formula syntax, model types, and some other stuff.

Value

A wbm_stan object, which is a list containing a model object with the brm model and a stan_code object with the model code.

If fit_model = FALSE, instead a list is returned containing a stan_code object and a stan_data object, leaving you with the tools you need to run the model yourself using rstan.

Author(s)

Jacob A. Long

See Also

wbm()

Examples

```
## Not run:
 data("WageData")
 wages <- panel_data(WageData, id = id, wave = t)
 model <- wbm_stan(lwage ~ lag(union) + wks | blk + fem | blk * lag(union),
           data = wages, chains = 1, iter = 2000)
 summary(model)

## End(Not run)
```

---

widen_panel *Convert long panel data to wide format*

Description

This function takes panel_data() objects as input as converts them to wide format for use in SEM and other situations when such a format is needed.

Usage

```
widen_panel(data, separator = "_", ignore.attributes = FALSE, varying = NULL)
```

Arguments

data	The panel_data frame.
separator	When the variables are labeled with the wave number, what should separate the variable name and wave number? By default, it is "_". In other words, a variable named var will be var_1, var_2, and so on in the wide data frame.
ignore.attributes	If the data was created by long_panel(), it stores information about which variables vary over time and which are constants. Sometimes, though, this information is not accurate (it is only based on the wide data's variable names) and you may want to force this function to check again based on the actual values of the variables.
varying	If you want to skip the checks for whether variables are varying and specify yourself, as is done with stats::reshape(), you can supply them as a vector here.

Details

This is a wrapper for `stats::reshape()`, which is renowned for being pretty confusing to use. This function automatically detects which of the variables vary over time and which don't, not appending wave information to constants.

Value

A data.frame with 1 row per respondent.

See Also

`reshape`

Examples

```
wages <- panel_data(WageData, id = id, wave = t)
wide_wages <- widen_panel(wages)
```

Index