Package 'plot3logit'

May 9, 2026

Type Package

Title Ternary Plots for Trinomial Regression Models

Version 3.2.0

Author Flavio Santi [cre, aut] (ORCID:
<https://orcid.org/0000-0002-2014-1981>),
Maria Michela Dickson [aut] (ORCID:
<https://orcid.org/0000-0002-4307-0469>),
Giuseppe Espa [aut] (ORCID: <https://orcid.org/0000-0002-0331-3630>),
Diego Giuliani [aut] (ORCID: <https://orcid.org/0000-0002-7198-6714>)

Maintainer Flavio Santi <flavio.santi@unitn.it>

URL https://www.flaviosanti.it/software/plot3logit/

BugReports https://github.com/f-santi/plot3logit/issues/

Description An implementation of the ternary plot for interpreting regression coefficients of trinomial regression models, as proposed in Santi, Dickson and Espa (2019) <doi:10.1080/00031305.2018.1442368>. Ternary plots can be drawn using either 'ggtern' package (based on 'ggplot2') or 'Ternary' package (based on standard graphics). The package and its features are illustrated in Santi, Dickson, Espa and Giuliani (2022) <doi:10.18637/jss.v103.c01>.

Depends R (>= 4.5), ggtern (>= 4.0.0), Ternary (>= 2.3.5)

Imports dplyr, ellipse, forcats, generics, ggplot2 (>= 4.0.1), graphics, grDevices, lifecycle, magrittr (>= 2.0.3), purrr, Rdpack, stats, stringr, tibble, tidyr, tidyselect, utils

Suggests knitr, MASS (>= 7.3.65), mlogit, nnet, ordinal, rmarkdown, VGAM, testthat

License GPL (>= 2)

LazyData true

NeedsCompilation no

Encoding UTF-8

RoxygenNote 7.3.2

Repository CRAN

VignetteBuilder knitr

RdMacros Rdpack, lifecycle

Date/Publication 2025-12-19 20:20:09 UTC

Contents

plot3logit-package . . . 2
add_confregions . . . 4
autoplot.Hfield3logit . . . 6
cross_1year . . . 7
extract3logit . . . 8
field3logit . . . 10
gg3logit . . . 14
labels . . . 15
multifield3logit . . . 16
stat_3logit . . . 18
stat_conf3logit . . . 19
stat_field3logit . . . 21
TernaryField . . . 23
USvote2016 . . . 24

Index **25**

`plot3logit-package` *Ternary Plots for Trinomial Regression Models*

Description

An implementation of the ternary plot for interpreting regression coefficients of trinomial regression models, as proposed in Santi et al. (2019). For details on the features of the package, see Santi et al. (2022).

Details

The package permits the covariate effects of trinomial regression models to be represented graphically by means of a ternary plot. The aim of the plots is helping the interpretation of regression coefficients in terms of the effects that a change in regressors' values has on the probability distribution of the dependent variable. Such changes may involve either a single regressor, or a group of them (composite changes), and the package permits both cases to be represented in a user-friendly way. Methodological details are illustrated and discussed in Santi et al. (2019).

The package can read the results of **both categorical and ordinal trinomial logit** regression fitted by various functions (see `extract3logit()`) and creates a `field3logit` object which may be represented by means of functions `autoplot()` and `plot()`.

The `plot3logit` package inherits graphical classes and methods from the package `ggtern` (Hamilton and Ferry 2018) which, in turn, is based on the `ggplot2` package (Wickham 2016).

Graphical representation based on **standard graphics** is made available through the package `Ternary` (Smith 2017) by function `TernaryField()` and in particular by the method `plot()` of `field3logit` class.

Since version 2.0.0, `plot3logit` can also compute and draw confidence regions associated to the covariate effects. See the package vignettes:

- type `vignette("plot3logit-overview")` for an overview of the package
- type `vignette("plot3logit-main")` for a presentation of all features (the vignette is based on Santi et al. 2022)

and the help of function `stat_conf3logit()` for some examples.

Compatibility

Function `field3logit()` can read trinomial regression estimates from the output of the following functions:

- `clm` and `clm2` of package `ordinal` (ordinal logit regression);
- `mlogit` of package `mlogit` (logit regression);
- `multinom` of package `nnet` (logit regression);
- `polr` of package `MASS` (ordinal logit regression);
- `vgam` and `vglm` of package `VGAM` (logit regression).

Moreover, explicit estimates can be passed to `field3logit()`. See examples and functions `field3logit()` and `extract3logit()` for further details.

Author(s)

Maintainer: Flavio Santi <flavio.santi@unitn.it> (ORCID)

Authors:

- Maria Michela Dickson <mariamichela.dickson@unitn.it> (ORCID)
- Giuseppe Espa <giuseppe.espa@unitn.it> (ORCID)
- Diego Giuliani <diego.giuliani@unitn.it> (ORCID)

References

Hamilton NE, Ferry M (2018). "ggtern: Ternary Diagrams Using ggplot2." *Journal of Statistical Software, Code Snippets*, **87**(3), 1-17. doi:10.18637/jss.v087.c03.

Santi F, Dickson MM, Espa G (2019). "A graphical tool for interpreting regression coefficients of trinomial logit models." *The American Statistician*, **73**(2), 200-207. doi:10.1080/00031305.2018.1442368.

Santi F, Dickson MM, Espa G, Giuliani D (2022). "plot3logit: Ternary Plots for Interpreting Trinomial Regression Models." *Journal of Statistical Software, Code Snippets*, **103**(1), 1–27. doi:10.18637/jss.v103.c01.

Smith MR (2017). "Ternary: An R Package for Creating Ternary Plots." *Zenodo*.

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag. ISBN 978-3-319-24277-4.

See Also

field3logit(), gg3logit(), TernaryField().

Examples

```
data(cross_1year)

# Read from "nnet::multinom" (categorical logit)
library(nnet)
mod0 <- multinom(employment_sit ~ gender + finalgrade, data = cross_1year)
field0 <- field3logit(mod0, 'genderFemale')
gg3logit(field0) + stat_field3logit()

# Read from "MASS::polr" (ordinal logit)
library(MASS)
mydata <- cross_1year
mydata$finalgrade <- factor(mydata$finalgrade,
  c('Low', 'Average', 'High'), ordered = TRUE)
mod1 <- polr(finalgrade ~ gender + irregularity, data = mydata)
field1 <- field3logit(mod1, 'genderFemale')
gg3logit(field1) + stat_field3logit()

# Read from list
mod2 <- list(
  B = matrix(
    data = c(-2.05, 0.46, -2.46, 0.37),
    nrow = 2,
    dimnames = list(c('(Intercept)', 'genderFemale'))
  ),
  levels = c('Employed', 'Unemployed', 'Trainee')
)
field2 <- field3logit(mod2, c(0, 1))
gg3logit(field2) + stat_field3logit()
```

---

add_confregions *Compute the confidence regions of covariate effects*

Description

Given the confidence level, it computes the confidence regions of the effects for each arrow of the `field3logit` or `multifield3logit` object given in input. If the `field3logit` or `multifield3logit` object already contains the confidence regions, they will be updated if the value of `conf` is different.

Usage

```
add_confregions(x, conf = 0.95, npoints = 100)
```

Arguments

x	an object of class field3logit or multifield3logit.
conf	confidence level of the regions.
npoints	number of points of the borders of the regions.

Details

Given a reference probability distribution π_0 over the simplex $S = \{(\pi^{(1)}, \pi^{(2)}, \pi^{(3)}) \in [0,1]^3 \colon \pi^{(1)} + \pi^{(2)} + \pi^{(3)} = 1\}$, and a change $\Delta \in \mathbf{R}^k$ of covariate values, the confidence region of the probability distribution resulting from the covariate change Δ is computed by means of the Wald statistics (Severini 2000), which should satisfy the following condition (Wooldridge 2010):

$$(\delta - \hat{\delta})^\top [(I_2 \otimes \Delta)^\top \hat{\Xi} (I_2 \otimes \Delta)]^{-1} (\delta - \hat{\delta}) \leq \chi^2_2(1-\alpha)$$

where $\hat{\delta} = \hat{B}^\top \Delta \in \mathbf{R}^2$ is the point estimate of change of natural parameters associated to Δ, $\hat{B} = [\beta^{(2)}, \beta^{(3)}] \in \mathbf{R}^{k \times 2}$ is the matrix of point estimates of regression coefficients, I_2 is the identity matrix of order two, $\otimes$ is the Kronecker product, $\hat{\Xi} \in \mathbf{R}^{2k \times 2k}$ is the covariance matrix of $vec(\hat{B})$, and finally, $\chi^2_2(1-\alpha)$ is the $(1-\alpha)$ quantile of χ^2_2.

The set of points which satisfy the previous inequality with equal sign delimits the border of the confidence region for δ.

If we denote with $\mathcal{R}_\delta$ the set of points δ which satisfy the previous inequality, it is possible to obtain the confidence region of the effect of the covariate change Δ over the simplex S as follows:

$$\mathcal{R} = \{g^{\leftarrow}(g(\pi_0) + \delta) \colon \delta \in \mathcal{R}_\delta\} \subseteq S$$

where $g \colon S \to \mathbf{R}^2$ and $g^{\leftarrow} \colon \mathbf{R}^2 \to S$ are respectively the link function of the trinomial logit model and its inverse. They are defined as follows:

$$g(\pi) = g([\pi^{(1)}, \pi^{(2)}, \pi^{(3)}]^\top) = \left[\ln \frac{\pi^{(2)}}{\pi^{(1)}}\,, \quad \ln \frac{\pi^{(3)}}{\pi^{(1)}}\right]^\top$$

$$g^{\leftarrow}(\eta) = g^{\leftarrow}([\eta_2, \eta_3]^\top) = \left[\frac{1}{1 + e^{\eta_2} + e^{\eta_3}}\,, \quad \frac{e^{\eta_2}}{1 + e^{\eta_2} + e^{\eta_3}}\,, \quad \frac{e^{\eta_3}}{1 + e^{\eta_2} + e^{\eta_3}}\right]^\top .$$

For further details and notation see Santi et al. (2022) and Santi et al. (2019).

Value

Object of class field3logit or multifield3logit with updated confidence regions.

References

Santi F, Dickson MM, Espa G (2019). "A graphical tool for interpreting regression coefficients of trinomial logit models." *The American Statistician*, **73**(2), 200-207. doi:10.1080/00031305.2018.1442368.

Santi F, Dickson MM, Espa G, Giuliani D (2022). "plot3logit: Ternary Plots for Interpreting Trinomial Regression Models." *Journal of Statistical Software, Code Snippets*, **103**(1), 1–27. doi:10.18637/jss.v103.c01.

Severini TA (2000). *Likelihood Methods in Statistics*. Oxford University Press. ISBN 978-0-19-850650-8.

Wooldridge JM (2010). *Econometric Analysis of Cross Section and Panel Data*, 2 edition. The MIT Press. ISBN 978-0-262-23258-6.

Examples

```
data(cross_1year)

mod0 <- nnet::multinom(employment_sit ~ gender + finalgrade,
  data = cross_1year)
field0 <- field3logit(mod0, 'genderFemale')
field0
add_confregions(field0)
```

---

autoplot.Hfield3logit *Create a* gg3logit *plot with field and confidence regions*

Description

`autoplot()` creates a `gg3logit` plot and adds a field and its confidence regions. `autoplot()` is a wrapper for `gg3logit()` and `stat_3logit()`.

Usage

```
## S3 method for class 'Hfield3logit'
autoplot(
  object,
  ...,
  mapping_field = aes(),
  mapping_conf = aes(),
  data = NULL,
  params_field = list(),
  params_conf = list(),
  show.legend = NA,
  conf = TRUE
)
```

Arguments

object	an object of class field3logit or multifield3logit.
...	other arguments passed to specific methods
mapping_field, mapping_conf	aesthetic mappings passed to argument mapping of stat_field3logit() and stat_conf3logit().
data	a field3logit object, a multifield3logit object, or a data.frame structured like a fortified field3logit or a multifield3logit object.
params_field, params_conf	graphical parameters passed to argument mapping of stat_field3logit() and stat_conf3logit().
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use TRUE. If NA, all levels are shown in legend, but unobserved levels are omitted.
conf	if TRUE and if confidence regions are available, the layer of stat_conf3logit() is added, otherwise only a gg3logit() object with the layer of stat_field3logit() is returned.

Value

Object of class ggplot.

See Also

Other gg functions: gg3logit(), stat_3logit(), stat_conf3logit(), stat_field3logit()

Examples

```
data(cross_1year)

mod0 <- nnet::multinom(employment_sit ~ gender + finalgrade, data = cross_1year)
field0 <- field3logit(mod0, 'genderFemale', conf = 0.95)

autoplot(field0)
```

---

cross_1year *Master's students' employment condition*

Description

data.frame with 3282 cross-sectional observations of 7 variables about employment condition of master's students one year after graduation. Data are used in Santi et al. (2019) and refer to students graduated at the University of Trento (Italy) between 2009 and 2013.

Format

`data.frame` with 3282 observations of 7 variables:

employment_sit: employment situation, a factor with three levels: *Employed*, *Unemployed*, *Trainee*.

gender: gender, a factor with two levels: *Male*, *Female*.

finalgrade: final grade degree, a factor with three levels: *Low*, *Average*, *High*.

duration: duration of studies, a factor with three levels: *Short*, *Average*, *Long*.

social_class: social class, a factor with five levels: *Working class*, *White-collar workers*, *Lower middle class*, *Upper middle class*, *Unclassified*.

irregularity: irregularity indicator of student's studies, a factor with three levels: *Low*, *Average*, *High*.

hsscore: high school final score, a numeric between 60 and 100.

References

Santi F, Dickson MM, Espa G (2019). "A graphical tool for interpreting regression coefficients of trinomial logit models." *The American Statistician*, **73**(2), 200-207. doi:10.1080/00031305.2018.1442368.

`extract3logit` *Extract information from fitted models*

Description

`extract3logit()` extracts all information which is relevant for computing the vector field(s) from the object passed to argument `x`. See **Details** for information on how new `S3` methods of generic `extract3logit()` should be implemented.

Usage

```
extract3logit(x, ...)

## Default S3 method:
extract3logit(x, ...)

## S3 method for class 'clm'
extract3logit(x, ...)

## S3 method for class 'clm2'
extract3logit(x, ...)

## S3 method for class 'mlogit'
extract3logit(x, ...)

## S3 method for class 'model3logit'
extract3logit(x, ...)
```

```
## S3 method for class 'multinom'
extract3logit(x, ...)

## S3 method for class 'polr'
extract3logit(x, ...)

## S3 method for class 'vgam'
extract3logit(x, ...)

## S3 method for class 'vglm'
extract3logit(x, ...)
```

Arguments

x	an object of any of the classes listed above. If instead, a list is passed, it should be structured as described in section **Details**.
...	other arguments passed to other methods.

Details

When a specific method is not available for a fitted model, it is possible to pass a list to argument x. In that case, the list should consists of the following components (the order is irrelevant):

- levels: vector of possible values of the dependent variable. It should be a character vector of lenght three, whose first element is interpreted as the reference level, whereas the second and the third elements are associated to the first and second columns of matrix B respectively.
- B: matrix of regression coefficients. It should be a numeric matrix (or any coercible object) with two columns if the model is cardinal, with only one column if the model is ordinal. The number of rows should be equal to the number of covariates and the names of covariates should be added as row names. The intercepts should be included only in case of categorical models, whereas column names, if provided, are ignored.
- alpha: intercepts of ordinal models. It should be a numerical vector of length two if the model is ordinal, otherwise this component should be either set to NULL or missing.
- vcovB: covariance matrix of regression coefficients. It should be a numeric matrix (or any coercible object) where the number of rows and columns equals the number of elements of B. Rows and columns should be ordered according to the labels of the dependent variable (slower index), and then to the covariates (faster index).

If a new S3 method for generic extract3logit() has to be implemented, the following components may be set:

- readfrom: character with information about the function that returned the estimates in the form package::function (for example nnet::multinom, MASS::polr, ...)

In any case, once the list has been created, the new method should invoke the default method extract3logit.default() and return its ouput. By so doing, automatic checks and initialisations are run before the model3logit object is returned.

Value

An object of class `model3logit`.

See Also

`plot3logit-package`, `field3logit()`.

`field3logit` *Computation of the vector field*

Description

`field3logit()` computes the vector field associated to a change in regressior values (which may involve more than one regressor) of a trinomial logit model either fitted by some multinomial regression function or explicitly specified.

The method `plot()` draws the ternary plot using standard graphics methods provided by package `Ternary`. See functions `gg3logit()` and `autoplot()` for plotting through the package `ggtern` based on the grammar of graphics.

Methods `as.data.frame()`, `as_tibble()`, fortify() and `tidy()` permits the graphical information of a `field3logit` object to be exported in a standardised format (either a `data.frame` or a `tibble`).

See Santi et al. (2022) for details and examples.

Usage

```
field3logit(
  model,
  delta,
  label = "",
  p0 = NULL,
  nstreams = 8,
  narrows = Inf,
  edge = 0.01,
  conf = NA,
  npoints = 100,
  alpha = deprecated(),
  vcov = deprecated()
)

## S3 method for class 'field3logit'
print(x, ...)

## S3 method for class 'field3logit'
plot(x, ..., add = FALSE, length = 0.05)

## S3 method for class 'field3logit'
as_tibble(x, ..., wide = TRUE)
```

```
## S3 method for class 'field3logit'
as.data.frame(x, ..., wide = TRUE)

## S3 method for class 'field3logit'
fortify(model, data, ..., wide = TRUE)

## S3 method for class 'field3logit'
tidy(x, ..., wide = TRUE)

## S3 method for class 'field3logit'
coef(object, ...)

## S3 method for class 'field3logit'
vcov(object, ...)

## S3 method for class 'field3logit'
labels(object, ...)

## S3 replacement method for class 'field3logit'
labels(object) <- value
```

Arguments

model	either a fitted trinomial model or a list properly structured. See section **Details** of `extract3logit()` and the last example of `plot3logit-package`.
delta	the change in the values of covariates to be represented. This could be either a `numeric` vector, the name of a covariate (passed either as a `character` or an `expression`), or a mathematical expression involving one or more than one covariates (passed either as a `character` or an `expression`). If a list is passed to `delta`, multiple fields are computed according to parameters passed as components of a 2-level list. See details and examples.
label	label to be used for identifying the field when multiple fields are plotted. See `multifield3logit()`.
p0	`list` of starting points (ternary coordinates) of the curves of the field. If not specified, `field3logit` automatically compute `nstreams` candidate points so that arrows are evenly distributed over the ternary plot area. See Examples.
nstreams	number of stream lines of the field to be computed. In case of ordinal models, this parameter is ineffective, as only one curve can be drawn. The parameter is ineffective also if argument `p0` is set.
narrows	maximum number of arrows to be drawn per stream line.
edge	minimum distance between each arrow (or point) and the edge of the ternary plot.
conf	confidence level of confidence regions to be computed **for each arrow** of the vector field.
npoints	number of points of the border to be computed **for each confidence region**.

alpha	deprecated argument. It may be removed in a future version of the package.
vcov	deprecated argument. It may be removed in a future version of the package.
x, object	object of class field3logit.
...	other arguments passed to or from other methods.
add	logical argument which specifies whether the field should be added to an existing plot (add = TRUE) or a new ternary plot should be drawn (add = FALSE).
length	length of the edges of the arrow head (in inches).
wide	it allows to choose whether as.data.frame, as_tibble, fortify and tidy should return a data.frame or a tibble in wide (default) or long form.
data	not used. Argument included only for interface compatibility with the generic fortify.
value	value to be assigned.

Details

The content of this section is presented with plenty of details and examples in Sections 4.1 and 4.3 of Santi et al. (2022).

Argument delta could be passed in one of the following formats:

- explicitly, as a numeric vector corresponding to the change $\Delta x \in \boldsymbol{R}^k$ in regressors values $x \in \boldsymbol{R}^k$;
- implicitly, as a character of the name of the covariate to be considered. In this case, vector $\Delta x \in \boldsymbol{R}^k$ is computed for a unit change of the specified covariate;
- as a mathematical expression (passed as an expression or a character object) involving one or more than one covariates. This allows one to analyse the effects of composite covariate changes through an easy-to-write and easy-to-read code without having to cope with explicit numerical specification of vector $\Delta x \in \boldsymbol{R}^k$.

See examples for comparing all three methods.

It is also possible to pass a list **to argument** delta**.** In such a case, the function field3logit is run once for every component of delta, and the set of generated field3logit objects is combined into a single object of class multifield3logit. The compoments of the list passed to delta must be named lists whose elements are used as arguments of each call of function field3logit, whilst the arguments specified in the parent call of field3logit are used as default values. It follows that arguments shared by all fields can be specified once in the parent call of field3logit, and only arguments which changes from field to field (such as delta and label) should be set in the lists making up the list passed to delta. See the penultimate example in section Examples and the help of multifield3logit().

Finally, when argument delta is a character, it is possible to indicate the name of a factor covariate between delimiters <<, >>. In that case, field3logit() creates a multifield3logit object where each field corresponds to the effect of each dummy generated by the factor regressor. If more than one regressor is included between delimiters <<, >>, all combinations between dummies are generated. See the last example in section **Examples**.

Value

S3 object of class `field3logit` structured as a named `list` or an object of class `multifield3logit` if `delta` is a `list` or syntax <<...>> is used.

See Also

`multifield3logit()`, `gg3logit()`, `autoplot()`.

Examples

```
data(cross_1year)


# Fitting the model
mod0 <- nnet::multinom(employment_sit ~ finalgrade + irregularity + hsscore,
  cross_1year)
mod0

# Assessing the effect of "finalgradeHigh" (explicit notation)
field0 <- field3logit(mod0, c(0, 0, 1, 0, 0, 0))
gg3logit(field0) + stat_field3logit()

# Assessing the effect of "finalgradeHigh" (implicit notation)
field0 <- field3logit(mod0, 'finalgradeHigh')
gg3logit(field0) + stat_field3logit()

# Assessing the combined effect of "finalgradeHigh" and
# a decrease of "hsscore" by 10
field0 <- field3logit(mod0, 'finalgradeHigh - 10 * hsscore')
gg3logit(field0) + stat_field3logit()


# Fitting the model
mod1 <- nnet::multinom(employment_sit ~ ., data = cross_1year)

# List passed to argument "delta" for generating "multifield3logit" objects
refpoint <- list(c(0.7, 0.15, 0.15))
depo <- list(
  list(delta = 'durationShort',  label = 'Short duration'),
  list(delta = 'durationLong',   label = 'Long duration'),
  list(delta = 'finalgradeHigh', label = 'High final grade'),
  list(delta = 'finalgradeLow',  label = 'Low final grade')
)
mfields <- field3logit(mod1, delta = depo, p0 = refpoint, narrows = 1)
mfields

# Sintax "<<...>>" for categorical covariates
mfields <- field3logit(
  model = mod1, delta = '<<finalgrade>>', label = 'Final grade',
  p0 = refpoint, narrows = 1
)
mfields
```

gg3logit *Create a new gg3logit*

Description

gg3logit initialises a ggplot object through ggtern. If a field3logit or a multifield3logit object is passed to argument data, the mandatory aesthetics of the ternary plot are automatically set. See Santi et al. (2022) for details and examples.

Usage

```
gg3logit(data = NULL, mapping = aes(), ...)
```

Arguments

data	a field3logit object, a multifield3logit object, or a data.frame structured like a fortified field3logit or a multifield3logit object.
mapping	list of aesthetic mappings to be used for plot. If a field3logit or a multifield3logit is passed to data, none of the aesthetics mappings listed in section *Aesthetic mappings* below has to be specified (if specified, they will be overwritten).
...	additional arguments passed through to ggtern.

Value

Object of class ggplot.

Aesthetic mappings

The following aesthetics are required by at least one of the available stats. None of them should be specified if a field3logit or a multifield3logit is passed to the argument data of gg3logit(), stat_field3logit() or stat_conf3logit():

- x, y, z are required by:
  - stat_field3logit() as ternary coordinates of the starting points of the arrows;
  - stat_conf3logit() ternary coordinates of the points on the border of confidence regions;
- xend, yend, zend: required by stat_field3logit() as ternary coordinates of the ending points of the arrows;
- group: identifier of groups of graphical objects (arrows and their confidence regions);
- type: type of graphical object (arrows or confidence regions).

The following variables of a fortified field3logit or a multifield3logit object may be useful for defining other standard aesthetics (such as fill, colour, ...):

- `label` identifies a field through a label, thus it is useful for distinguishing the fields in a `multifield3logit` object.
- `idarrow` identifies each group of graphical objects (arrows and their confidence regions) *within* every field. Unlike variable `group`, `idarrow` is not a global identifier of graphical objects.

See Also

Other gg functions: `autoplot.Hfield3logit()`, `stat_3logit()`, `stat_conf3logit()`, `stat_field3logit()`

Examples

```
data(cross_1year)

mod0 <- nnet::multinom(employment_sit ~ gender + finalgrade, data = cross_1year)
field0 <- field3logit(mod0, 'genderFemale')

gg3logit(field0) + stat_field3logit()
```

---

labels *Set the labels of a* `field3logit` *or a* `multifield3logit` *object*

Description

It sets the labels of an existing `field3logit` or a `multifield3logit` object.

Usage

```
labels(object) <- value
```

Arguments

`object`	a `field3logit` or a `multifield3logit` object.
`value`	a character with the new label (or labels in case of a `multifield3logit` object).

Value

Object of same class of argument `object` (either `field3logit` or `multifield3logit`).

`multifield3logit` *Multiple trilogit fields*

Description

Methods of `S3` class `multifield3logit` handle multiple `fields3logit` objects simultaneously and permit new `multifield3logit` objects to be easily created by means of the sum operator "+".

Usage

```
multifield3logit(x, ...)

## S3 method for class 'Hfield3logit'
x + y

## S3 method for class 'multifield3logit'
print(x, maxitems = 10, ...)

## S3 method for class 'multifield3logit'
plot(x, y = NULL, add = FALSE, col = NA, legend = TRUE, ...)

## S3 method for class 'multifield3logit'
as_tibble(x, ..., wide = TRUE)

## S3 method for class 'multifield3logit'
as.data.frame(x, ..., wide = TRUE)

## S3 method for class 'multifield3logit'
fortify(model, data, ..., wide = TRUE)

## S3 method for class 'multifield3logit'
tidy(x, ..., wide = TRUE)

## S3 method for class 'multifield3logit'
labels(object, ...)

## S3 replacement method for class 'multifield3logit'
labels(object) <- value

## S3 method for class 'multifield3logit'
x[i, drop = TRUE]

## S3 replacement method for class 'multifield3logit'
x[i] <- value
```

Arguments

x, y, model, object	object of class field3logit or multifield3logit.
...	other arguments passed to or from other methods.
maxitems	maximum number of items to be enumerated when an object of class multifield3logit is printed.
add	logical argument which specifies whether the field should be added to an existing plot (add = TRUE) or a new ternary plot should be drawn (add = FALSE).
col, legend	graphical parameters if Ternary package is used.
wide	it allows to choose whether as.data.frame, as_tibble, fortify and tidy should return a data.frame or a tibble in wide (default) or long form.
data	not used. Argument included only for interface compatibility with the generic fortify.
value	value to be assigned.
i	index of the field3logit object to be selected.
drop	if TRUE, a field3logit object is returned if the subsetted multifield3logit object has length one.

Value

S3 object of class multifield3logit structured as a named list.

See Also

field3logit().

Examples

```
data(cross_1year)

mod0 <- nnet::multinom(employment_sit ~ ., data = cross_1year)
mod0

field_Sdur <- field3logit(mod0, 'durationShort',
  label = 'Short duration')
field_Hfgr <- field3logit(mod0, 'finalgradeHigh',
  label = 'High final grade')

gg3logit(field_Sdur + field_Hfgr) +
  stat_field3logit()
  facet_wrap(~ label)

refpoint <- list(c(0.7, 0.15, 0.15))

field_Sdur <- field3logit(mod0, 'durationShort',
  label = 'Short duration', p0 = refpoint, narrows = 1)
field_Ldur <- field3logit(mod0, 'durationLong',
  label = 'Long duration', p0 = refpoint, narrows = 1)
```

```
field_Hfgr <- field3logit(mod0, 'finalgradeHigh',
  label = 'High final grade', p0 = refpoint, narrows = 1)
field_Lfgr <- field3logit(mod0, 'finalgradeLow',
  label = 'Low final grade', p0 = refpoint, narrows = 1)

mfields <- field_Sdur + field_Ldur  + field_Lfgr + field_Hfgr
mfields

gg3logit(mfields) +
  stat_field3logit(aes(colour = label)) +
  theme_zoom_L(0.45)
```

`stat_3logit` *Add a field and confidence regions to a* `gg3logit` *plot*

Description

`stat_3logit()` adds a field and confidence regions to a `gg3logit` plot. `stat_3logit()` is a wrapper for stats `stat_field3logit()` and `stat_conf3logit()` which are jointly applied.

Usage

```
stat_3logit(
  mapping_field = aes(),
  mapping_conf = aes(),
  data = NULL,
  params_field = list(),
  params_conf = list(),
  show.legend = NA,
  inherit.aes = TRUE,
  conf = TRUE
)
```

Arguments

`mapping_field, mapping_conf`
: aesthetic mappings passed to argument `mapping` of `stat_field3logit()` and `stat_conf3logit()`.

`data`
: a `field3logit` object, a `multifield3logit` object, or a `data.frame` structured like a fortified `field3logit` or a `multifield3logit` object.

`params_field, params_conf`
: graphical parameters passed to argument `mapping` of `stat_field3logit()` and `stat_conf3logit()`.

show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use TRUE. If NA, all levels are shown in legend, but unobserved levels are omitted.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. annotation_borders().
conf	if TRUE and if confidence regions are available, the layer of stat_conf3logit() is added, otherwise only the layer of stat_field3logit() is returned.

Value

If conf is set to FALSE a layer of ggplot package is returned (object of class LayerInstance), otherwise, if conf is set to TRUE, stat_3logit returns a list of two ggplot2 layers (class LayerInstance).

See Also

Other gg functions: autoplot.Hfield3logit(), gg3logit(), stat_conf3logit(), stat_field3logit()

Examples

```
data(cross_1year)

mod0 <- nnet::multinom(employment_sit ~ gender + finalgrade, data = cross_1year)
field0 <- field3logit(mod0, 'genderFemale', conf = 0.95)

gg3logit(field0) + stat_3logit()
gg3logit(field0) + stat_3logit(conf = TRUE)
```

---

stat_conf3logit *Add the confidence regions of a field to a* gg3logit *plot*

Description

stat_conf3logit() adds a field to a gg3logit plot.

Usage

```
stat_conf3logit(
  mapping = aes(),
  data = NULL,
  geom = "polygon",
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
```

```
  ...
)
```

Arguments

mapping	list of aesthetic mappings to be used for plot. Mandatory aesthetics should not be specified if field3loglit or multifield3logit object is passed to data. See secion **Aesthetic mappings** of gg3logit() for details.
data	a field3logit object, a multifield3logit object, or a data.frame structured like a fortified field3logit or a multifield3logit object.
geom	The geometric object to use to display the data for this layer. When using a stat_*() function to construct a layer, the geom argument can be used to override the default coupling between stats and geoms. The geom argument accepts the following: • A Geom ggproto subclass, for example GeomPoint. • A string naming the geom. To give the geom as a string, strip the function name of the geom_ prefix. For example, to use geom_point(), give the geom as "point". • For more information and other ways to specify the geom, see the layer geom documentation.
position	A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following: • The result of calling a position function, such as position_jitter(). This method allows for passing extra arguments to the position. • A string naming the position adjustment. To give the position as a string, strip the function name of the position_ prefix. For example, to use position_jitter(), give the position as "jitter". • For more information and other ways to specify the position, see the layer position documentation.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use TRUE. If NA, all levels are shown in legend, but unobserved levels are omitted.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. annotation_borders().
...	additional arguments passed through to ggtern.

Value

Layer of ggplot2 package, object of class LayerInstance.

See Also

Other gg functions: autoplot.Hfield3logit(), gg3logit(), stat_3logit(), stat_field3logit()

Examples

```
data(cross_1year)

mod0 <- nnet::multinom(employment_sit ~ gender + finalgrade, data = cross_1year)
field0 <- field3logit(mod0, 'genderFemale', conf = 0.95)

gg3logit(field0) + stat_conf3logit()
gg3logit(field0) + stat_field3logit() + stat_conf3logit()
```

---

stat_field3logit *Add a field to a* gg3logit *plot*

Description

stat_field3logit() adds a field to a gg3logit plot.

Usage

```
stat_field3logit(
  mapping = aes(),
  data = NULL,
  geom = "segment",
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
  arrow. = arrow(length = unit(0.2, "cm")),
  ...
)
```

Arguments

mapping	list of aesthetic mappings to be used for plot. Mandatory aesthetics should not be specified if field3loglit or multifield3logit object is passed to data. See secion **Aesthetic mappings** of gg3logit() for details.
data	a field3logit object, a multifield3logit object, or a data.frame structured like a fortified field3logit or a multifield3logit object.
geom	The geometric object to use to display the data for this layer. When using a stat_*() function to construct a layer, the geom argument can be used to override the default coupling between stats and geoms. The geom argument accepts the following: • A Geom ggproto subclass, for example GeomPoint. • A string naming the geom. To give the geom as a string, strip the function name of the geom_ prefix. For example, to use geom_point(), give the geom as "point".

- For more information and other ways to specify the geom, see the layer geom documentation.

`position` A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The `position` argument accepts the following:

- The result of calling a position function, such as `position_jitter()`. This method allows for passing extra arguments to the position.
- A string naming the position adjustment. To give the position as a string, strip the function name of the `position_` prefix. For example, to use `position_jitter()`, give the position as `"jitter"`.
- For more information and other ways to specify the position, see the layer position documentation.

`show.legend` logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use `TRUE`. If `NA`, all levels are shown in legend, but unobserved levels are omitted.

`inherit.aes` If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. `annotation_borders()`.

`arrow.` specification for arrow heads, as created by function `arrow` of package `grid`.

`...` additional arguments passed through to `ggtern`.

Value

Layer of `ggplot2` package, object of class `LayerInstance`.

See Also

Other gg functions: `autoplot.Hfield3logit()`, `gg3logit()`, `stat_3logit()`, `stat_conf3logit()`

Examples

```
data(cross_1year)

mod0 <- nnet::multinom(employment_sit ~ gender + finalgrade, data = cross_1year)
field0 <- field3logit(mod0, 'genderFemale', conf = 0.95)

gg3logit(field0) + stat_field3logit()
gg3logit(field0) + stat_field3logit() + stat_conf3logit()
```

`TernaryField` *Draw a field on an existing ternary plot*

Description

`TernaryField()` adds the vector field returned by `field3logit()` to an existing ternary plot generated by `Ternary::TernaryPlot()`.

Usage

```
TernaryField(
  field,
  ...,
  length = 0.05,
  conf = FALSE,
  npoints = 100,
  conf.args = list()
)
```

Arguments

`field`	object of class `field3logit` as returned by `field3logit()`.
`...`	other arguments passed to or from other methods.
`length`	length of the edges of the arrow head (in inches).
`conf`	if `FALSE` confidence regions are not drawn, even if available; if `TRUE` confidence regions are drawn only if available; if a `numeric` value is passed, confidence regions at the specified confidence level are computed (if not already available) and drawn.
`npoints`	number of points of the border to be computed **for each confidence region**.
`conf.args`	graphical parameters of confidence regions to be passed to `Ternary::TernaryPolygon()`.

Value

An object of class `field3logit` with confidence regions included, if computed within `TernaryField()`.

See Also

`field3logit()`.

Examples

```
library(nnet)
data(cross_1year)

mod0 <- nnet::multinom(employment_sit ~ gender + finalgrade, data = cross_1year)
field0 <- field3logit(mod0, 'genderFemale')
```

```
TernaryPlot()
TernaryField(field0)
```

`USvote2016` *Self-reported votes from VOTER Survey in 2016*

Description

Dataset based on self-reported votes from 2016 VOTER Survey by Democracy Fund Voter Study Group (2017), as used in the examples in Santi et al. (2022).

Format

`tibble` (`data.frame`) with 8000 observations of 7 variables:

idcode: voter identifier (integer).

vote: declared vote, afactor with three levels: *Clinton*, *Trump*, *Other*.

race: race, a factor with six levels: *White*, *Black*, *Hispanic*, *Asian*, *Mixed*, *Other*.

educ: level of education, a factor with six levels: *No high school*, *High school grad.*, *Some college*, *2-year college*, *4-year college*, *Post-grad*.

gender: gender, a factor with four levels: *Male*, *Female*, *Skipped*, *Not Asked*.

birthyr: decades when the voter was born, a factor with six levels: *[1920,1940)*, *[1940,1950)*, *[1950,1960)*, *[1960,1970)*, *[1970,1980)*, *[1980,2000)*.

famincome: income (in USD) of voter's family, a factor with five levels: *[0; 30,000)*, *[30,000; 60,000)*, *[60,000; 100,000)*, *[100,000; 150,000)*, *[150,000; Inf)*.

Details

Object `USvote2016` includes only few variables based on the result of the survey, which are publicly available online. See file `"data-raw/USvote2016_prepare.R"` in the GitHub repository `"f-santi/plot3logit"` (https://github.com/f-santi/plot3logit), where it is documented how the dataset `USvote2016` has been generated.

References

Democracy Fund Voter Study Group (2017). "Views of the electorate research survey, December 2016." https://www.voterstudygroup.org.

Santi F, Dickson MM, Espa G, Giuliani D (2022). "plot3logit: Ternary Plots for Interpreting Trinomial Regression Models." *Journal of Statistical Software, Code Snippets*, **103**(1), 1–27. doi:10.18637/jss.v103.c01.

Index

∗ **data**
    cross_1year, 7
    USvote2016, 24
∗ **gg functions**
    autoplot.Hfield3logit, 6
    gg3logit, 14
    stat_3logit, 18
    stat_conf3logit, 19
    stat_field3logit, 21
+.Hfield3logit *(*multifield3logit*)*, 16
[.multifield3logit *(*multifield3logit*)*, 16
[<-.multifield3logit *(*multifield3logit*)*, 16

add_confregions, 4
annotation_borders(), *19, 20*, *22*
arrow, *22*
as.data.frame(), *10*
as.data.frame.field3logit *(*field3logit*)*, 10
as.data.frame.multifield3logit *(*multifield3logit*)*, 16
as_tibble(), *10*
as_tibble.field3logit *(*field3logit*)*, 10
as_tibble.multifield3logit *(*multifield3logit*)*, 16
autoplot(), *2*, *6*, *10*, *13*
autoplot.Hfield3logit, 6, *15*, *19, 20*, *22*

clm, *3*
clm2, *3*
coef.field3logit *(*field3logit*)*, 10
cross_1year, 7

extract3logit, 8
extract3logit(), *2, 3*, *8, 9*, *11*
extract3logit.default(), *9*

field3logit, 10
field3logit(), *3, 4*, *10*, *12*, *17*, *23*
fortify(), *10*
fortify.field3logit *(*field3logit*)*, 10
fortify.multifield3logit *(*multifield3logit*)*, 16

gg3logit, *6, 7*, 14, *18–22*
gg3logit(), *4*, *6, 7*, *10*, *13, 14*, *20, 21*
ggplot, *14*
ggplot2, *2*
ggtern, *2*, *10*, *14*, *20*, *22*
grid, *22*

labels, 15
labels.field3logit *(*field3logit*)*, 10
labels.multifield3logit *(*multifield3logit*)*, 16
labels<- *(*labels*)*, 15
labels<-.field3logit *(*field3logit*)*, 10
labels<-.multifield3logit *(*multifield3logit*)*, 16
layer geom, *20*, *22*
layer position, *20*, *22*

mlogit, *3*
multifield3logit, 16
multifield3logit(), *11–13*
multinom, *3*

plot(), *2, 3*, *10*
plot.field3logit *(*field3logit*)*, 10
plot.multifield3logit *(*multifield3logit*)*, 16
plot3logit *(*plot3logit-package*)*, 2
plot3logit-package, 2
polr, *3*
print.field3logit *(*field3logit*)*, 10
print.multifield3logit *(*multifield3logit*)*, 16

stat_3logit, *7*, *15*, 18, *20*, *22*

stat_3logit(), *6*, *18*
stat_conf3logit, *7*, *15*, *19*, 19, *22*
stat_conf3logit(), *3*, *7*, *14*, *18, 19*
stat_field3logit, *7*, *15*, *19, 20*, 21
stat_field3logit(), *7*, *14*, *18, 19*, *21*

Ternary::TernaryPlot(), *23*
Ternary::TernaryPolygon(), *23*
TernaryField, 23
TernaryField(), *3, 4*, *23*
tidy(), *10*
tidy.field3logit *(*field3logit*)*, 10
tidy.multifield3logit
        *(*multifield3logit*)*, 16

USvote2016, 24

vcov.field3logit *(*field3logit*)*, 10
vgam, *3*
vglm, *3*