

Package ‘plot3logit’

September 8, 2019

Type Package

Title Ternary Plots for Trinomial Regression Models

Version 1.0.2

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

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

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

BugReports <https://github.com/f-santi/plot3logit>

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](https://doi.org/10.1080/00031305.2018.1442368)>. Ternary plots can be drawn using either 'ggtern' package (based on 'ggplot2') or 'Ternary' package (based on standard graphics).

Depends R (>= 3.1), ggtern (>= 3.1.0), Ternary (>= 1.0.1)

Imports ellipse, ggplot2 (>= 3.1.0), graphics, magrittr (>= 1.5),
reshape2 (>= 1.4.3), stats, utils

Suggests MASS (>= 7.3-51), mlogit, nnet, knitr, rmarkdown

License GPL (>= 2)

LazyData true

NeedsCompilation no

Encoding UTF-8

RoxygenNote 6.1.1

Repository CRAN

Date 2019-09-08

VignetteBuilder knitr

Date/Publication 2019-09-08 15:10:02 UTC

R topics documented:

plot3logit-package	2
cross_1year	4
field3logit	4
gg3logit	7
multifield3logit	8
stat_3logit	9
TernaryField	10

Index	12
--------------	-----------

plot3logit-package	<i>Ternary Plots for Trinomial Regression Models</i>
--------------------	--

Description

An implementation of the ternary plot for interpreting regression coefficients of trinomial regression models, as proposed in Santi, Dickson and Espa (2019).

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. Theoretical and methodological details are illustrated and discussed in Santi, Dickson and Espa (2019).

The package can read the results of **both categorical and ordinal trinomial logit** regression fitted by various functions (see the next section) and creates a `field3logit` object which may be represented by means of functions `gg3logit` and `stat_3logit`.

The `plot3logit` package inherits graphical classes and methods from the package `ggtern` (Hamilton 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 functions `plot3logit` and `TernaryField`, and in particular by the `plot` method of `field3logit` objects.

Compatibility

Function `field3logit` of package `plot3logit` can read trinomial regression estimates from the output of the following functions:

- `multinom` of package `nnet` (logit regression);
- `polr` of package `MASS` (ordinal logit regression);
- `mlogit` of package `mlogit` (logit regression).

Moreover, explicit matrix of regression coefficients can be passed to `field3logit`. See examples and function `field3logit` for further details.

References

- Hamilton N. E., M. Ferry (2018) "ggtern: Ternary Diagrams Using ggplot2", *Journal of Statistical Software, Code Snippets*, **87**(3), 1-17. doi: [10.18637/jss.v087.c03](https://doi.org/10.18637/jss.v087.c03)
- Santi F., M. M. Dickson, G. Espa (2019) "A graphical tool for interpreting regression coefficients of trinomial logit models", *The American Statistician*, **73**(2), pp. 200-207. doi: [10.1080/00031305.2018.1442368](https://doi.org/10.1080/00031305.2018.1442368)
- Smith M. R. (2017). "Ternary: An R Package for Creating Ternary Plots", *Zenodo*. doi: [10.5281/zenodo.1068996](https://doi.org/10.5281/zenodo.1068996)
- Wickham, H. (2016) *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag, New York.

See Also

[field3logit](#), [gg3logit](#), [TernaryField](#).

Examples

```
data(cross_1year)

# Read from "nnet::multinom"
library(nnet)
mod0 <- multinom(employment_sit ~ gender + finalgrade, data = cross_1year)
field0 <- field3logit(mod0, 'genderFemale')
gg3logit(field0) + stat_3logit()

# Read from "MASS::polr"
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_3logit()

# Read from "mlogit::mlogit"
library(mlogit)
mydata <- mlogit.data(cross_1year, choice = 'employment_sit', shape = 'wide')
mod2 <- mlogit(employment_sit ~ 0 | gender + finalgrade, data = mydata)
field2 <- field3logit(mod2, 'genderFemale')
gg3logit(field2) + stat_3logit()

# Read from matrix
M <- matrix(c(-2.05, 0.46, -2.46, 0.37), nrow = 2)
rownames(M) <- c('(Intercept)', 'genderFemale')
attr(M, 'labs') <- c('Employed', 'Unemployed', 'Trainee')
field3 <- field3logit(M, c(0, 1))
gg3logit(field3) + stat_3logit()
```

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 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., M. M. Dickson, G. Espa (2019) "A graphical tool for interpreting regression coefficients of trinomial logit models", *The American Statistician*, **73**(2), pp. 200-207. doi: [10.1080/00031305.2018.1442368](https://doi.org/10.1080/00031305.2018.1442368)

field3logit

Computation of the vector field

Description

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

plot3logit and plot method draw the ternary plot using standard graphics methods provided by package Ternary. See function `gg3logit` for plotting through the package `ggtern` based on the grammar of graphics.

Usage

```

field3logit(model, delta, label = "<empty>", p0 = NULL, alpha = NULL,
  ncurves = 8, narrows = Inf, edge = 0.01)

plot3logit(model, delta, label = "<empty>", p0 = NULL, alpha = NULL,
  ncurves = 8, narrows = Inf, edge = 0.01, ...)

## 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.data.frame(x, ..., wide = TRUE)

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

```

Arguments

model	either a fitted trinomial model or a matrix of regressor coefficients. See section <i>Compatibility</i> and examples 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). 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 ncurves candidate points so that arrows are evenly distributed over the ternary plot area. See Examples.
alpha	numeric vector of length two where constants $\alpha^{(1)}$ and $\alpha^{(2)}$ are stored (only for ordinal models), as defined in Equation (7) of Santi, Dickson and Espa (2019).
ncurves	number of curves 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 in case that argument p0 is set.
narrows	maximum number of arrows to be drawn per curve.
edge	minimum distance between each arrow (or point) and the edge of the ternary plot.
...	other arguments passed to or from other methods.
x	object of class field3logit.
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 <code>as.data.frame</code> should return a <code>data.frame</code> object in wide (default) or long form.
data	not used. Argument included only for interface compatibility with the generic <code>fortify</code> .

Details

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

- explicitly, as a numeric vector corresponding to the change $\Delta x \in \mathbf{R}^k$ in regressors values $x \in \mathbf{R}^k$;
- implicitly, as a character of the name of the covariate to be considered. In this case, vector $\Delta x \in \mathbf{R}^k$ 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 $\Delta x \in \mathbf{R}^k$.

See examples for comparing all three methods.

Value

S3 object of class `field3logit` structured as a named list.

References

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

See Also

[multifield3logit](#), [gg3logit](#).

Examples

```
data(cross_1year)

# Model fit
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_3logit()

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

```
gg3logit(field0) + stat_3logit()

# Assessing the combined effect of "finalgradeHigh" and
# a decrease of "hsscore" by 10
field0 <- field3logit(mod0, 'finalgradeHigh - 10 * hsscore')
gg3logit(field0) + stat_3logit()
```

gg3logit

Create a new gg3logit

Description

gg3logit initializes a [ggplot](#) object through [ggtern](#). If a fortified [field3logit](#) or a [multifield3logit](#) object is passed as argument data to gg3logit, the labels of the ternary plot are automatically. The same happens if a [field3logit](#) or a [multifield3logit](#) object is passed; in that case, gg3logit preliminarily invoke the fortify method.

Usage

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

Arguments

data	a field3logit or a multifield3logit object. If not specified, must be supplied in each layer added to the plot.
mapping	list of aesthetic mappings to use for plot. If not specified, must be supplied in each layer added to the plot. Note that mappings x, y and z are not required: they will be overwritten if specified (see examples).
...	additional arguments passed through to ggtern .

See Also

Other 'gg' functions: [multifield3logit](#), [stat_3logit](#)

Examples

```
data(cross_1year)

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

gg3logit(field0) + stat_3logit()
```

multifield3logit *Multiple trilogit fields*

Description

Methods of S3 class `multifield3logit` handle multiple `field3logit` objects simultaneously and permit new `multifield3logit` object to be easily created by means of the sum operator `+`. The `fortify` method of `multifield3logit` permits multiple `field3logit` objects to be easily handled by `gg3logit` and ggtern-based functions and methods. See Examples.

Usage

```
multifield3logit(x, ...)

## S3 method for class 'field3logit'
x + y

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

## S3 method for class 'multifield3logit'
fortify(model, data, ...)
```

Arguments

<code>x, y</code>	object of class <code>field3logit</code> or <code>multifield3logit</code> .
<code>...</code>	other arguments passed to or from other methods.
<code>maxitems</code>	maximum number of items to be enumerated when an object of class <code>multifield3logit</code> is printed.
<code>model</code>	either a fitted trinomial model or a matrix of regressor coefficients. See section <i>Compatibility</i> and examples of plot3logit-package .
<code>data</code>	not used. Argument included only for interface compatibility with the generic <code>fortify</code> .

Value

S3 object of class `multifield3logit` structured as a named list.

See Also

[field3logit](#).

Other ‘gg’ functions: [gg3logit](#), [stat_3logit](#)

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_3logit() +
  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_3logit(aes(colour = label)) +
  theme_zoom_L(0.45)
```

stat_3logit

Add a field to a gg3logit plot

Description

stat_3logit add a field to a [gg3logit](#) plot.

Usage

```
stat_3logit(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 use for plot. Note that mappings x, y and z are not required: they will be overwritten if specified (see examples).
data	a <code>field3logit</code> or a <code>multifield3logit</code> object.
geom	The geometric object to use display the data
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
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.
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. <code>borders()</code> .
arrow.	specification for arrow heads, as created by function <code>arrow</code> of package <code>grid</code> .
...	Other arguments passed on to <code>layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired <code>geom/stat</code> .

See Also

Other 'gg' functions: [gg3logit](#), [multifield3logit](#)

Examples

```
data(cross_1year)

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

gg3logit(field0) + stat_3logit()
gg3logit() + stat_3logit(data = field0)
```

TernaryField

Draw a field on an existing ternary plot

Description

TernaryField adds to an existing ternary plot generated by [TernaryPlot](#) the vector field returned by [field3logit](#).

Usage

```
TernaryField(field, ..., length = 0.05)
```

Arguments

`field` object of class `field3logit` as returned by `field3logit`.
`...` [graphical parameters](#) such as `xpd` and the line characteristics `lend`, `ljoin` and `lmitre`: see `par`.
`length` length of the edges of the arrow head (in inches).

Value

Argument `field` is invisibly returned.

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)
```

Index

*Topic **data**
 cross_1year, 4
+.field3logit (multifield3logit), 8
arrow, 10
as.data.frame.field3logit
 (field3logit), 4

borders(), 10

cross_1year, 4

field3logit, 2, 3, 4, 8, 10, 11
fortify, 8
fortify.field3logit (field3logit), 4
fortify.multifield3logit
 (multifield3logit), 8

gg3logit, 2–4, 6, 7, 8–10
ggplot, 7
ggplot2, 2
ggtern, 2, 4, 7
graphical parameters, 11
grid, 10

layer(), 10

mlogit, 2
multifield3logit, 5–7, 8, 10
multinom, 2

par, 11
plot, 2
plot.field3logit (field3logit), 4
plot3logit, 2
plot3logit (field3logit), 4
plot3logit-package, 2
polr, 2
print.field3logit (field3logit), 4
print.multifield3logit
 (multifield3logit), 8

stat_3logit, 2, 7, 8, 9

Ternary, 2
TernaryField, 2, 3, 10
TernaryPlot, 10