Package ‘lba’

February 20, 2015

Title Latent Budget Analysis for Compositional Data

Version 1.0

Date 2015-01-22

Author Enio G. Jelihovschi <eniojelihovs@gmail.com>
       Ivan Bezerra Allaman <ivanalaman@gmail.com>

Maintainer Enio G. Jelihovschi <eniojelihovs@gmail.com>

Depends R (>= 3.1.2), MASS, alabama, plotrix, ca

Description Latent budget analysis is a method for the analysis of a two-way contingency table with an exploratory variable and a response variable. It is specially designed for compositional data.

Encoding latin1

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2015-01-22 18:42:44

R topics documented:

  goodnessfit .......................................................... 2
  Latent Budget Analysis ............................................. 5
  lba ................................................................. 5
  MANHATAN .......................................................... 13
  PerfMark ......................................................... 14
  plot.lba .......................................................... 16
  print.goodnessfit ................................................... 19
  summary.lba .................................................... 20
  votB ............................................................... 21

Index 23
goodnessfit

*Goodness of Fit results for Latent Budget Analysis*

**Description**

The goodness of fit results assesses how well the model fits the data. It consists of measures of the resemblance between the observed and the expected data, and the parsimony of the model.

**Usage**

`goodnessfit(object)`

**Arguments**

- `object` An object of one of following classes: `lba.ls`, `lba.ls.fe`, `lba.ls.logit`, `lba.mle`, `lba.mle.fe`, `lba.mle.logit`

**Value**

The `goodnessfit` function to the method `lba.mle`, `lba.mle.fe` and `lba.mle.logit` returns a list with the slots:

- `dfdb` Degrees of freedom of the base model
- `dfd` Degrees of freedom of the full model
- `G2b` Likelihood ratio statistic of the base model
- `G2` Likelihood ratio statistic of the full model
- `chi2b` Chi-square statistic of the base model
- `chi2` Chi-square statistic of the full model
- `proG1` P-value of likelihood ratio statistic of the base model
- `proG` P-value of likelihood ratio statistic of the full model
- `prochi1` P-value of chi-square statistic of the base model
- `prochi` P-value of chi-square statistic of the full model
- `AICb` AIC criteria of the base model
- `AICC` AIC criteria of the full model
- `BICb` BIC criteria of the base model
- `BICC` BIC criteria of the full model
- `CAICb` CAIC criteria of the base model
- `CAIC` CAIC criteria of the full model
- `delta1` Normed fit index
- `delta2` Normed fit index modified
- `rho1` Bollen index
The `goodnessfit` function to the method `lba.1s`, `lba.1s.fe` and `lba.1s.logit` returns a list with the slots:

- **rho2**: Tucker-Lewis index
- **RSS1**: Residual sum of square of the base model
- **RSS**: Residual sum of square of the full model
- **impRSS**: Improvement of RSS
- **impPB**: Improvement per budget
- **impDF**: Average improvement per degree of freedom
- **D1**: Index of dissimilarity of the base model
- **D**: Index of dissimilarity of the full model
- **pccb**: Proportion of correctly classified data of the base model
- **pcc**: Proportion of correctly classified data of the full model
- **impD**: Improvement of proportion of correctly classified data
- **impPCCB**: Improvement of Proportion of correctly classified data per budget
- **AmpPCCDF**: Average improvement of Proportion of correctly classified data per degree of freedom
- **mad1**: Mean angular deviation of the base model
- **madk**: Mean angular deviation of the full model
- **impMad**: Improvement mean angular deviation
- **impPBasat**: Improvement mean angular deviation per budget
- **impDFsat**: Average improvement mean angular deviation per degree of freedom
- **dfdb**: Degrees of freedom of the base model
- **dfd**: Degrees of freedom of the full model
- **RSS1**: Residual sum of square of the base model
- **RSS**: Residual sum of square of the full model
- **impRSS**: Improvement of RSS
- **impPB**: Improvement per budget
- **impDF**: Average improvement per degree of freedom
- **wRSS1**: Weighted residual sum of square of the base model
- **wRSS**: Weighted residual sum of square of the full model
- **impwRSS**: Improvement of wRSS
- **D1**: Index of dissimilarity of the base model
- **D**: Index of dissimilarity of the full model
- **pccb**: Proportion of correctly classified data of the base model
- **pcc**: Proportion of correctly classified data of the full model
- **impD**: Improvement of proportion of correctly classified data
- **impPCCB**: Improvement of Proportion of correctly classified data per budget
AimppCCDF  Average improvement of Proportion of correctly classified data per degree of freedom
mad1     Mean angular deviation of the base model
madk     Mean angular deviation of the full model
impMad   Improvement mean angular deviation
impPBSat Improvement mean angular deviation per budget
impDFsat Average improvement mean angular deviation per degree of freedom

Note
For a detailed and complete discussion about goodness of fit results for latent budget analysis, see van der Ark 1999.

References

See Also
print.goodnessfit, lba

Examples

data('votB')

# Using LS method (default) without constraint
# K = 2
ex1 <- lba(city ~ parties,
           votB,
           K = 2)

gx1 <- goodnessfit(ex1)
gx1

# Using MLE method without constraint
# K = 2
exm <- lba(city ~ parties,
           votB,
           K = 2,
           method='mle')

gxm <- goodnessfit(exm)
gxm
Latent Budget Analysis

Latent Budget Analysis (LBA) for Compositional Data

Description

Latent budget analysis (LBA) is a method for the analysis of contingency tables, from where the compositional data is derived. It is used to understand the relationship between the table rows and columns, where the rows denote the categories of the explanatory variable and the columns denote the categories of the response variable.

Details

The row vectors of the compositional data are called observed budgets which are approximated by the expected budgets. The LBA allows us to find which categories of the response are related to different groups of the explanatory categories. If the table has a product multinomial distribution we can understand the latent budget model (LBM) as explaining the relationship between the explanatory and the response variables assuming that conditioned on the latent variable they are independent. In that sense, the latent budgets, which are categories of a latent variable, are hidden values which explain the relationship between the explanatory and response variables. LBA reduce the dimensionality of the original problem, thus making it easier to understand its hidden relations.

Author(s)

Enio G. Jelihovschi <eniojelihovs@gmail.com> and Ivan Bezerra Allaman <ivanalaman@gmail.com>. We are grateful to José C. Faria, Andreas L. van der Ark and Fábio M. Correa for suggestions.

Maintainer: Enio G. Jelihovschi <eniojelihovs@gmail.com>

Usage

lba

## S3 method for class 'matrix'
lba(obj, 
       A = NULL, 

lbaHobjL NNNI 
CC sS method for class G 

lbaHobjL 
a = nullL
B = NULL,
K = 1L,
cA = NULL,
cB = NULL,
logitA = NULL,
logitB = NULL,
omsk = NULL,
psitk = NULL,
S = NULL,
T = NULL,
row.weights = NULL,
col.weights = NULL,
tolG = 1e-10,
tolA = 1e-05,
tolB = 1e-05,
itmax.unide = 1e3,
itmax.ide = 1e3,
trace.lba = TRUE,
toltype = "all",
method = c("ls", "mle"),
what = c("inner","outer"), ...)

## S3 method for class 'table'

lba(obj,
    A = NULL,
    B = NULL,
    K = 1L,
    cA = NULL,
    cB = NULL,
    logitA = NULL,
    logitB = NULL,
    omsk = NULL,
    psitk = NULL,
    S = NULL,
    T = NULL,
    row.weights = NULL,
    col.weights = NULL,
    tolg = 1e-10,
    tola = 1e-05,
    tolb = 1e-05,
    itmax.unide = 1e3,
    itmax.ide = 1e3,
    trace.lba = TRUE,
    toltype = "all",
    method = c("ls", "mle"),
    what = c("inner","outer"), ...
)

## S3 method for class 'formula'
lba(formula, data,
    A = NULL,
    B = NULL,
    K = 1L,
    cA = NULL,
    cB = NULL,
    logitA = NULL,
    logitB = NULL,
    omsk = NULL,
    psitk = NULL,
    S = NULL,
    T = NULL,
    row.weights = NULL,
    col.weights = NULL,
    tolG = 1e-10,
    tolA = 1e-05,
    tolB = 1e-05,
    itmax.unide = 1e3,
    itmax.ide = 1e3,
    trace.lba = TRUE,
    toltype = "all",
    method = c("ls", "mle"),
    what = c("inner","outer"), ...)

## S3 method for class 'ls'
lba(obj,
    A ,
    B ,
    K ,
    row.weights ,
    col.weights ,
    tolA ,
    tolB ,
    itmax.unide ,
    itmax.ide ,
    trace.lba ,
    what , ...)

## S3 method for class 'mle'
lba(obj,
    A ,
    B ,
    K ,
    tolG ,
    tolA ,
    tolB ,
    itmax.unide ,
    itmax.ide ,
    trace.lba ,
    what , ...)

## S3 method for class 'lba'
lba(obj,
    A ,
    B ,
    K ,
    row.weights ,
    col.weights ,
    tolA ,
    tolB ,
    itmax.unide ,
    itmax.ide ,
    trace.lba ,
    what , ...)

lba(method = c("ls", "mle"),
    what = c("inner", "outer"),
    toltype = "all",
    method = c("ls", "mle"),
    what = c("inner", "outer"), ...)

## S3 method for class 'lm'
lba(obj,
    A ,
    B ,
    K ,
    row.weights ,
    col.weights ,
    tolA ,
    tolB ,
    itmax.unide ,
    itmax.ide ,
    trace.lba ,
    what , ...)

## S3 method for class 'glm'
lba(obj,
    A ,
    B ,
    K ,
    row.weights ,
    col.weights ,
    tolA ,
    tolB ,
    itmax.unide ,
    itmax.ide ,
    trace.lba ,
    what , ...)

## S3 method for class 'glm'
lba(formula, data,
    g = NULL,
    ghat = NULL,
    A = NULL,
    B = NULL,
    K = 1L,
    cA = NULL,
    cB = NULL,
    logitA = NULL,
    logitB = NULL,
    omsk = NULL,
    psitk = NULL,
    S = NULL,
    T = NULL,
    row.weights = NULL,
    col.weights = NULL,
    tolG = 1e-10,
    tolA = 1e-05,
    tolB = 1e-05,
    itmax.unide = 1e3,
    itmax.ide = 1e3,
    trace.lba = TRUE,
    toltype = "all",
    method = c("ls", "mle"),
    what = c("inner", "outer"), ...)

## S3 method for class 'glm'
lba(obj,
    g = NULL,
    ghat = NULL,
    A ,
    B ,
    K ,
    row.weights ,
    col.weights ,
    tolA ,
    tolB ,
    itmax.unide ,
    itmax.ide ,
    trace.lba ,
    what , ...)

## S3 method for class 'lba'
lba(method = c("ls", "mle"),
    what = c("inner", "outer"),
    toltype = "all",
    method = c("ls", "mle"),
    what = c("inner", "outer"), ...)

## S3 method for class 'lm'
lba(obj,
    g = NULL,
    ghat = NULL,
    A ,
    B ,
    K ,
    row.weights ,
    col.weights ,
    tolA ,
    tolB ,
    itmax.unide ,
    itmax.ide ,
    trace.lba ,
    what , ...)

## S3 method for class 'glm'
lba(formula, data,
    g = NULL,
    ghat = NULL,
    A = NULL,
    B = NULL,
    K = 1L,
    cA = NULL,
    cB = NULL,
    logitA = NULL,
    logitB = NULL,
    omsk = NULL,
    psitk = NULL,
    S = NULL,
    T = NULL,
    row.weights = NULL,
    col.weights = NULL,
    tolG = 1e-10,
    tolA = 1e-05,
    tolB = 1e-05,
    itmax.unide = 1e3,
    itmax.ide = 1e3,
    trace.lba = TRUE,
    toltype = "all",
    method = c("ls", "mle"),
    what = c("inner", "outer"), ...)

## S3 method for class 'lm'
lba(obj,
    g = NULL,
    ghat = NULL,
    A ,
    B ,
    K ,
    row.weights ,
    col.weights ,
    tolA ,
    tolB ,
    itmax.unide ,
    itmax.ide ,
    trace.lba ,
    what , ...)

## S3 method for class 'glm'
lba(formula, data,
    g = NULL,
    ghat = NULL,
    A = NULL,
    B = NULL,
    K = 1L,
    cA = NULL,
    cB = NULL,
    logitA = NULL,
    logitB = NULL,
    omsk = NULL,
    psitk = NULL,
    S = NULL,
    T = NULL,
    row.weights = NULL,
    col.weights = NULL,
    tolG = 1e-10,
    tolA = 1e-05,
    tolB = 1e-05,
    itmax.unide = 1e3,
    itmax.ide = 1e3,
    trace.lba = TRUE,
    toltype = "all",
    method = c("ls", "mle"),
    what = c("inner", "outer"), ...)

## S3 method for class 'lm'
lba(obj,
    g = NULL,
    ghat = NULL,
    A ,
    B ,
    K ,
    row.weights ,
    col.weights ,
    tolA ,
    tolB ,
    itmax.unide ,
    itmax.ide ,
    trace.lba ,
    what , ...)

## S3 method for class 'glm'
lba(formula, data,
    g = NULL,
    ghat = NULL,
    A = NULL,
    B = NULL,
    K = 1L,
    cA = NULL,
    cB = NULL,
    logitA = NULL,
    logitB = NULL,
    omsk = NULL,
    psitk = NULL,
    S = NULL,
    T = NULL,
trace.lba ,
toltype ,
what , ...)

## S3 method for class 'ls.fe'
lba(obj,
    A ,
    B ,
    K ,
    cA ,
    cB ,
    row.weights ,
    col.weights ,
    itmax.ide ,
    trace.lba , ...)

## S3 method for class 'mle.fe'
lba(obj,
    A ,
    B ,
    K ,
    cA ,
    cB ,
tolG ,
tolA ,
tolB ,
    itmax.ide ,
    trace.lba ,
toltype , ...)

## S3 method for class 'ls.logit'
lba(obj,
    A ,
    B ,
    K ,
    cA ,
    cB ,
    logitA ,
    logitB ,
    omsk ,
    psitk ,
    S ,
    T ,
    row.weights ,
    col.weights ,
    itmax.ide ,
    trace.lba , ...)

## S3 method for class 'mle.logit'

```r
lba(obj,
    A ,
    B ,
    K ,
    cA ,
    cB ,
    logitA ,
    logitB ,
    omsk ,
    psitk ,
    S ,
    T ,
    itermax.idx,
    trace.lba , ...)
```

### Arguments

- **obj,formula**
  The function is generic, accepting some forms of the principal argument for specifying a two-way frequency table. Currently accepted forms are matrix, data frame (coerced to frequency tables), objects of class "xtabs" or "table" and one-sided formulae of the form `Row1 + Row2 + ... + Rown ~ Col1 + Col2 + ... + Coln`, where `Rown` and `Coln` are `nth row` (the mixing parameters) and column variable (the latent components).

- **data**
  A data frame containing variables in `formula`.

- **A**
  The starting value of a (I x K) matrix containing the mixing parameters, if given. The default is NULL, producing random starting values.

- **B**
  The starting value of a (J x K) matrix containing the latent components, if given. The default is NULL, producing random starting values.

- **K**
  Integer giving the number of latent budgets chosen by the user. The default is 1.

- **cA**
  The value of a (I x K) matrix containing the constraints on the mixing parameters. Fixed constraints are the values themselves which are numbers in the [0,1] interval. The optional equality constraints are indicated by an integer starting from 2, such that parameters that must be equal have the same integer. The default is NULL, indicating no constraints.

- **cB**
  The value of a (J x K) matrix containing the constraints on the latent components. Fixed constraints are the values themselves which are numbers in the [0,1] interval. The optional equality constraints are indicated by an integer starting from 2, such that parameters that must be equal have the same integer. The default is NULL, indicating no constraints.

- **logitA**
  Design (I x S) matrix for row-covariates. The first column contains 1’s, indicating a constant covariate. The entries may be continuous or dummy coded values.

- **logitB**
  Design (J x T) matrix for column-covariates. The entries may be continuous or dummy coded values.

- **omsk**
  A (S x K) matrix giving the starting values for the multinomial logit parameters of the row covariates. The default is NULL, producing random starting values.
psitk  
A (TxK) matrix giving the starting values for the multinomial logit parameters of the column covariates. The default is NULL, producing random starting values.

S  
Number of row-covariates. The default is NULL.

T  
Number of column-covariates. The default is NULL.

row.weights  
Row weights for weighted least squares method. The default is NULL.

col.weights  
Column weights for weighted least squares method. The default is NULL. If both row.weights and col.weights are NULL and "1s" method is chosen, then ordinary least squares is used.

tolG  
A tolerance value for judging when convergence has been reached. It is based on the estimated likelihood ratio statistics $G^2$. The default is $1e^{-10}$.

tolA  
A tolerance value for judging when convergence has been reached. When the one-iteration change in the maximum of the absolute value of the element wise difference of the estimated matrices $A$ is less than tolA. The default is $1e^{-05}$.

tolB  
A tolerance value for judging when convergence has been reached. When the one-iteration change in the maximum of the absolute value of the element wise difference of the estimated matrices $B$ is less than tolB. The default is $1e^{-05}$.

itmax.unide  
Maximum number of iterations performed by the mle or ls method, if convergence is not achieved, before identification parameters. The default is $1e3$.

itmax.ide  
Maximum number of iterations performed by the mle or ls method in the identification process. Is used too when the constrained fixed, equality and logit are required. The default is $1e3$.

trace.lba  
Logical, indicating whether the base function optim and constrOptim.nl from package alabama, will trace their results. The default is TRUE.

toltype  
String indicating which kind of tolerance to be used. That is, the EM algorithm stops updating and considers the maximum log-likelihood to have been found. Their types are: "all" when the one-iteration change in the estimated likelihood ratio statistics $G^2$ is less than tolG, and the one-iteration change in the maximum of the absolute value of the element wise difference of the estimated matrices $A$ is less than tolA and the same for estimated matrices $B$ with respect to tolB; "$G2" when the only one-iteration change in the estimated likelihood ratio statistics $G^2$ is less than tolG; "ab" when only the one-iteration change in the maximum of the absolute value of the element wise difference of the estimated matrices $A$ is less than tolA and the same for estimated matrices $B$ with respect to tolB. toltype works only for method = "mle". The default is "all". The ls method uses only "ab" as tolerance limit.

method  
String indicating which kind of estimating method. They are: "1s" when least squares, either weighted or ordinary, method is used; "mle" when maximum likelihood method is used. The default is "1s".

what  
String indicating which kind identified solutions for mixing parameters and latent budgets matrices. They are: the "inner" extreme solution and the "outer" extreme solution. The default is "inner".

...  
Potential further arguments (required by generic).
The method `lba.ls` and `lba.mle` returns a list of class `lba.ls` and `lba.mle` respectively with the slots:

- **P**
  - The compositional data matrix which is formed by dividing the raw data matrix by their corresponding total, its rows are called observed budgets.

- **pij**
  - Matrix whose rows are the expected budgets.

- **residual**
  - Residual matrix $P - pij$.

- **A**
  - $(I \times K)$ matrix of the unidentified the mixing parameters.

- **B**
  - $(J \times K)$ matrix of the unidentified the latent components.

- **Aoi**
  - $(I \times K)$ matrix of the identified mixing parameters, they may be either the inner extreme values or the outer extreme values.

- **Boi**
  - $(J \times K)$ matrix of the identified latent components, they may be either the inner extreme values or the outer extreme values.

- **rescB**
  - $(J \times K)$ matrix of the rescaled latent components.

- **pk**
  - Budget proportions.

- **val_func**
  - Value of least squared or likelihood function achieved.

- **iter_unide**
  - Number of unidentified iterations.

- **iter_ide**
  - Number of identified iterations.

The method `lba.ls.fe` and `lba.mle.fe` returns a list of class `lba.ls.fe` and `lba.mle.fe` respectively with the slots:

- **P**
  - The compositional data matrix which is formed by dividing the raw data matrix by their corresponding total, its rows are called observed budgets.

- **pij**
  - Matrix whose rows are the expected budgets.

- **residual**
  - Residual matrix $P - pij$.

- **A**
  - $(I \times K)$ matrix of the unidentified the mixing parameters.

- **B**
  - $(J \times K)$ matrix of the unidentified the latent components.

- **pk**
  - Budget proportions.

- **val_func**
  - Value of least squared or likelihood function achieved.

- **iter_ide**
  - Number of identified iterations.

The method `lba.ls.logit` and `lba.mle.logit` returns a list of class `lba.ls.logit` and `lba.mle.logit` respectively with the slots:

- **P**
  - The compositional data matrix which is formed by dividing the raw data matrix by their corresponding total, its rows are called observed budgets.

- **pij**
  - Matrix whose rows are the expected budgets.

- **residual**
  - Residual matrix $P - pij$.

- **A**
  - $(I \times K)$ matrix of the unidentified the mixing parameters.

- **B**
  - $(J \times K)$ matrix of the unidentified the latent components.
Budget proportions.

Value of least squared or likelihood function achieved.

Number of identified iterations.

A \((S \times K)\) matrix giving estimated values of the multinomial logit parameters of the row covariates.

A \((T \times K)\) matrix giving the estimated values for the multinomial logit parameters of the column covariates.

Note

The user has two options to entry the data: the raw data and tabulated data. If the raw data is imported, he may indicate which, among the variables, comprises the row and which the column variable and let the `lba.formula` function make the tabulation. The user may also tabulate the data with the available functions in R. Recalling that if this second option is used, the object must be of the class `xtabs`, `table` or `matrix`. If the user imports the tabulated data, the class is, in general, `data.frame` and so, it is necessary to transform the object data into a `matrix`.

The function `lba` uses EM algorithm to maximise the latent budget model log-likelihood function; the Active Constraints Methods (ACM) to minimise either the weighted least squares (wls), or ordinary least squares (ols) functions; and "BFGS" variable metric method in `constrOptim.nl` function of `alabama` package and in `optim` function of `stats` package used in identification for \(K \geq 3\), in constraint algorithm for ls method, in multinomial logit constraints and in some parts of constraining for mle method. Depending on the starting parameters, those algorithms may only locate a local, rather than global, maximum. This becomes more and more of a problem as \(K\), the number of latent budgets, increases. It is therefore highly advisable to run `lba` multiple times until you are relatively certain that you have located the global maximum log-likelihood or the global minimum least squares.

References


See Also

goodnessfit, summary.lba, plot.lba
Examples

data('votB')

# Using LS method (default) without constraint
# K = 2
ex1 <- lba(city ~ parties,
    votB,
    K = 2)
ex1

# Already tabulated data? Ok!
data('PerfMark')
ex2 <- lba(as.matrix(PerfMark),
    K = 2,
    what='outer')
ex2

# Using LS method (default) with constraint
# Fixed constraint to mixing parameters
caki1 <- matrix(c(0.2, NA, NA,
    NA, NA, 0.2,
    NA, NA, 0.2,
    0.3, NA, NA,
    0.2, NA, NA,
    NA, NA, NA),
    byrow = TRUE,
    ncol = 3)

# K = 3
exf1 <- lba(city ~ parties,
    votB,
    cA = caki1,
    K = 3)
exf1

Description

The MANHATAN data frame has 25 rows and 3 columns. The observations were obtained in a study carried out by the sociologist Leo Srole and describe the cross-classification of 1660 adults in Manhattan, ages 20-59, obtained from a sample of midtown residents.

Usage

MANHATAN
**Format**

This data frame contains the following columns:

- **health** A factor with levels: Well; Misy; Mosy; Imp.
- **socecon** A factor with levels: A; B; C; D; E; F.
- **value** The absolute frequencies of which factor.

**Source**


**References**


---

**Description**

The PerfMark data frame has 31 rows and 46 columns. The data set is the result of a survey of 47 beauty salons located at the city of Lavras, Brazil, consisting of two types of questions; the first identifies the profile of the owner manager (explanatory variable), the second are questions referring to the degree of professionalism with respect to planning, market and finances (response variable). The data set is already cross-tabulated.

**Usage**

PerfMark

**Format**

This data frame contains the following columns referring the absolute frequencies to each row variable:

Planning variables:

- **PA14** What is the dependence of the owner to function properly?.
- **PA20** What are your plans towards next year? only a dream.
- **PA21** What are your plans towards next year? vague goals.

Marketing variables:

- **MA11** Your business tries to systematically assess the customer satisfaction and use that as a basis for management decisions. Alternative 1.
- **MA12** Your business tries to systematically assess the customer satisfaction and use that as a basis for management decisions. Alternative 2.
- **MA20** Your business offers more than the usual services. Alternative 0.
MA21 Your business offers more than the usual services. Alternative 1.
MA30 Your business is focused to further customer loyalty. Alternative 0.
MA31 Your business is focused to further customer loyalty. Alternative 1.
MA32 Your business is focused to further customer loyalty. Alternative 2.
MA42 What is the proportion, among current customers, of those who are customers for more than 6 months. Alternative 2.
MA43 What is the proportion, among current customers, of those who are customers for more than 6 months. Alternative 3.
MB12 Your business offers more services than when it began. Alternative 2.
MB22 How is your business quality perceived as compared to the competition? Alternative 2.
MB23 How is your business quality perceived as compared to the competition? Alternative 3.
MB31 How is your business range of services perceived as compared to the competition? Alternative 1.
MB32 How is your business range of services perceived as compared to the competition? Alternative 2.
MC11 What is your business level of prices perceived as compared to the competition? Alternative 1.
MC12 What is your business level of prices perceived as compared to the competition? Alternative 2.
MD13 Your business location is perceived as appropriate to the target market. Alternative 3.
ME10 Your business uses formal media to advertise itself. Alternative 0.
ME11 Your business uses formal media to advertise itself. Alternative 1.
ME25 Your business uses formal media to advertise itself. Alternative 5. Financial variables:
F10 Your business clearly separates the owner bills from the business bills. Alternative 0.
F14 Your business clearly separates the owner bills from the business bills. Alternative 4.
F20 Your owners withdrawal are planned and controlled in advance. Alternative 0.
F21 Your owners withdrawal are planned and controlled in advance. Alternative 1.
F24 Your owners withdrawal are planned and controlled in advance. Alternative 4.
F31 Your business pays for its purchases in installments. Alternative 1.
F34 Your business pays for its purchases in installments. Alternative 4.
F42 Your business knows today whether it will be able to pay its short-term bills of 60 days. Alternative 2.
F44 Your business knows today whether it will be able to pay its short-term bills of 60 days. Alternative 4.
F50 Your business uses short-term cash-flow analysis to plan for its short-term bills. Alternative 0.
F63 Your business has formal control of the monthly amount it makes from its services. Alternative 3.
F64 Your business has formal control of the monthly amount it makes from its services. Alternative 4.
F70 Your business uses either credit card, checkbook payment or loans, to finance its needs for working capital. Alternative 0.
F74 Your business uses either credit card, checkbook payment or loans, to finance its needs for working capital. Alternative 4.
F80 Your business uses specific credit to finance its needs for capital. Alternative 0.
F91 The company demonstrates knowledge to properly assess the costs of products used in services and costs of renting and taxes. Alternative 1.
F93 The company demonstrates knowledge to properly assess the costs of products used in services and costs of renting and taxes. Alternative 3.
F100 Your business clearly identifies the need for working capital. Alternative 0.
F111 Your business lays down the price of services in a systematic way. Alternative 1.
F113 Your business lays down the price of services in a systematic way. Alternative 3.
F120 The company calculates the interest on contracted loans. Alternative 0.
F125 The company calculates the interest on contracted loans. Alternative 5.

Source

References

---

**plot.lba**

*Plot lba objects*

**Description**

S3 methods for lba objects. It’s possible two types of visualisation: the lba type, suggested by van der Ark (1999) and correspondence analysis suggested by Jelihovschi (2011).

**Usage**

```r
## S3 method for class 'lba'
plot(x,
    budget.prop = TRUE,
    col.points = NULL,
    col.lines = NULL,# only to K = 2
    col.budget = NULL,
)```

Arguments

\begin{itemize}
\item \texttt{x} \hspace{1cm} A object of \texttt{lba} class.
\item \texttt{budget.prop} \hspace{0.5cm} A line representing the budget proportion. The default is \texttt{TRUE}.
\item \texttt{col.points} \hspace{0.5cm} A vector of colours representing the mixing parameters and latent components. The default is \texttt{NULL}.
\item \texttt{col.lines} \hspace{0.5cm} A vector of colours representing the mixing parameters and latent components. Only to \texttt{K = 2}. The default is \texttt{NULL}.
\item \texttt{col.budget} \hspace{0.5cm} A vector of colour representing the budget proportion. The default is \texttt{NULL}.
\item \texttt{pch.points} \hspace{0.5cm} A vector of plotting characters or symbols representing the mixing parameters and latent components. The default is \texttt{NULL}.
\item \texttt{pch.budget} \hspace{0.5cm} A vector of plotting characters or symbols representing the budget proportion. The default is \texttt{NULL}.
\item \texttt{lty.lines} \hspace{0.5cm} A vector of line types representing the mixing parameters and latent components. Only to \texttt{K = 2}. The default is \texttt{NULL}.
\item \texttt{lty.budget} \hspace{0.5cm} A vector of line types representing the budget proportion. The default is \texttt{NULL}.
\item \texttt{lwd.lines} \hspace{0.5cm} A vector of line width representing the mixing parameters and latent components. The default is \texttt{NULL}.
\item \texttt{lwd.budget} \hspace{0.5cm} A vector of line width representing the budget proportion. The default is \texttt{NULL}.
\item \texttt{legend} \hspace{0.5cm} A logical indicating whether a legend should be included. The default is \texttt{NULL}.
\item \texttt{with.ml} \hspace{0.5cm} What's parameters do you like to plot? The default is mixing parameters ("mix").
\item \texttt{type} \hspace{0.5cm} The type of options graphical. The default is "lba". See details.
\item \ldots \hspace{0.5cm} Optional plotting parameters.
\end{itemize}

Details

\texttt{plot.lba} plots two types of graphics. The first, \texttt{(type = 'lba')}, is the one suggested at de Leeuw et all (1990) and at van der Ark (1999) thesis. In this type, it is only possible the graphical views for \texttt{K = 2} and \texttt{K = 3}. When \texttt{K = 2}, the heads of the latent budgets are be connected by a (one dimensional) line segment. When \texttt{K = 3}, the heads of the latent budgets are the vertices of a triangle, and the plot is made with help of \texttt{triax.plot} function of \texttt{plotrix} package. In the second type, \texttt{plot.ca}, suggested at Jelihovschii et all (2011), the graphical display is made by making use of correspondence analysis graphics of the mixing parameters and latent components matrices. This is done with the function \texttt{ca} of \texttt{ca} package. In this case, a graphic display is possible for \texttt{K >= 3}. 
Author(s)

Enio G. Jelihovschi (<eniojelihovs@gmail.com>)
Ivan Bezerra Allaman (<ivanalaman@gmail.com>)

References


See Also

*triax.plot, triax.points, plot.ca.*

Examples

data('votb')
ex1 <- lba(city ~ parties,
data=votb,
   K = 2)
plot(ex1)

# It's very simple. with colors!
plot(ex1,
   col.points=1:6,
   col.lines=1:6)

# I don't want your output. I want to provide my values and to use only your graphics. Ok!!!!
# Example: pag. 34, inner extreme solution
k1 <- c(.16,.68,.06,.45,.54,.30,.00,.45,1,.19,.62,.68,.57,.50,.30,
   .88,.46,.69,.62,.26,.53,.47,.58,.09,.26,.40,.70,.16,.71,.32)

k2 <- c(.84,.32,.94,.55,.46,.7,1,.55,.00,.81,.38,.32,.43,.50,.7,
   .12,.54,.91,.38,.74,.47,.53,.42,.91,.74,.60,.30,.84,.29,.68)

my_mixing <- cbind(k1,k2)
rownames(my_mixing) <- paste(rep(c('H','C','R','Q','T','N','Se','Sk','L','V'),
   rep(3,10)),
   c('*','=','-'),
   sep = '')

my_budget <- matrix(c(.43,.57),
   ncol=2)

colnames(my_budget) <- paste('LB',1:2,sep='')

my_data <- list(NA,
   NA,
NA,
NA,
NA,
my_mixing,
NA,
NA,
my_budget)
class(my_data) = 'lba'

# I didn't like. I want equal of book. Pag. 41.

l_points <- c(2,24,17,
               rep(3,3),
               6,6,25,
               rep(4,3),
               1,16,21,
               rep(11,3),
               5,5,23,
               rep(8,3),
               rep(9,3),
               rep(13,3))

my_colors <- rep(c('red','blue','black'),
                  10)

plot(my_data,
     pch.points = l_points,
     col.points = my_colors,
     lty.budget = 2,
     pch.budget = 7,
     legend = FALSE)
legend(-0.1,
       1.2,
       rownames(my_mixing),
       pch = l_points,
       col = my_colors,
       xpd = TRUE,
       cex = 0.8,
       ncol = 10) # Is beautiful!!

print.goodnessfit

Print Method for goodnessfit objects.

Description

Returns (and prints) a summary list for goodnessfit objects.

Usage

## S3 method for class 'goodnessfit'
print(x, digits = 2L, ...)

summary.lba

Arguments

- `x`: A given object of the class `goodnessfit.lba.ls` and `goodnessfit.lba.mle`.
- `digits`: Number of decimal digits in the results.
- `...`: Potential further arguments (require by generic).

Author(s)

Enio G. Jelihovschi (<eniojelihovs@gmail.com>)
Ivan Bezerra Allaman (<ivanalaman@gmail.com>)

See Also

`goodnessfit`

Examples

```r
data('votB')

# Using LS method (default) without constraint
# K = 2
ex1 <- lba(city ~ parties,
    votB,
    K = 2)
exm <- goodnessfit(ex1)
exm
```

summary.lba  

**Summary Method for lba objects.**

Description

Returns (and prints) a summary list for objects of class `lba`, `lba.ls.fe`, `lba.mle.fe`, `lba.ls.logit` and `lba.mle.logit`.

Usage

```r
## S3 method for class 'lba'
support(object, digits = 2L, ...)
```

Arguments

- `object`: A given object of the class `lba`, `lba.ls.fe`, `lba.mle.fe`, `lba.ls.logit` and `lba.mle.logit`.
- `digits`: Number of decimal digits in the results.
- `...`: Potential further arguments (require by generic).
Author(s)

Enio G. Jelihovschi (<eniojelihovs@gmail.com>)
Ivan Bezerra Allaman (<ivanalaman@gmail.com>)

See Also

lba

Examples

data('votB')

# Using LS method (default) without constraint
# K = 2
ex1 <- lba(city ~ parties,
            votB, 
            K = 2)
summary(ex1)

---

votB Voting Behaviour in Netherlands

Description

The votB data frame has 8971 rows and 2 columns. The raw data refers to the type of the city and the political party which each participant voted for in the 1986 general elections in the Netherlands.

Usage

votB

Format

This data frame contains the following columns:

- **city** A factor with levels: co Commuter; lx Large city; mc Middle large city; ri Rural industrialised; ru Rural; sc Small city.
- **parties** A factor with levels: cda Christian democrats; dVV Democrats; left Other left-wing parties; Pvda Labor party; right Other right-wing parties; vvd Liberals.

Source

References

Index

*Topic correspondence analysis
  plot.lba, 16
*Topic datasets
  MANHATAN, 13
  PerfMark, 14
*Topic dataset
  votB, 21
*Topic goodnessfit
  print.goodnessfit, 19
*Topic lba
  plot.lba, 16
  summary.lba, 20
*Topic package
  Latent Budget Analysis, 5
  print.goodnessfit, 19
  summary.lba, 20
*Topic plot
  plot.lba, 16
  goodnessfit, 2, 12, 20

  Latent Budget Analysis, 5
  lba, 4, 5, 21
  lba-package (Latent Budget Analysis), 5

  MANHATAN, 13
  PerfMark, 14
  plot.ca, 18
  plot.lba, 12, 16
  print.goodnessfit, 4, 19

  summary.lba, 12, 20

  triax.plot, 18
  triax.points, 18

  votB, 21