Package 'aldvmm'

July 22, 2025

Type Package

Title Adjusted Limited Dependent Variable Mixture Models

Version 0.8.8

Date 2023-10-26

Depends R (>= 3.5.0)

Description The goal of the package 'aldvmm' is to fit adjusted limited dependent variable mixture models of health state utilities. Adjusted limited dependent variable mixture models are finite mixtures of normal distributions with an accumulation of density mass at the limits, and a gap between 100% quality of life and the next smaller utility value. The package 'aldvmm' uses the likelihood and expected value functions proposed by Hernandez Alava and Wailoo (2015) <doi:10.1177/1536867X1501500307> using normal component distributions and a multinomial logit model of probabilities of component membership.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports numDeriv, stats, checkmate, optimx, Formula, sandwich, lmtest

URL https://github.com/pletschm/aldvmm/

BugReports https://github.com/pletschm/aldvmm/issues

Suggests knitr, kableExtra, markdown, tinytex, testthat (>= 3.0.0), covr, rmarkdown, bookdown, xtable, ggplot2, scales, reshape2

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Mark Pletscher [aut, cre, cph], Achim Zeileis [ctb] (ORCID: <https://orcid.org/0000-0003-0918-3766>)

Maintainer Mark Pletscher <pletscher.mark@gmail.com>

Repository CRAN

Date/Publication 2023-10-28 05:40:08 UTC

Contents

`aldvmm-package` *aldvmm: Adjusted Limited Dependent Variable Mixture Models*

Description

The goal of the package 'aldvmm' is to fit adjusted limited dependent variable mixture models of health state utilities. Adjusted limited dependent variable mixture models are finite mixtures of normal distributions with an accumulation of density mass at the limits, and a gap between 100% quality of life and the next smaller utility value. The package 'aldvmm' uses the likelihood and expected value functions proposed by Hernandez Alava and Wailoo (2015) <doi:10.1177/1536867X1501500307> using normal component distributions and a multinomial logit model of probabilities of component membership.

Examples

```
data(utility)

 fit <- aldvmm(eq5d ~ age + female | 1,
               data = utility,
               psi = c(0.883, -0.594),
               ncmp = 2)

 summary(fit)

 yhat <- predict(fit)
```

---

aldvmm *Fitting Adjusted Limited Dependent Variable Mixture Models*

Description

The function `aldvmm()` fits adjusted limited dependent variable mixture models of health state utilities. Adjusted limited dependent variable mixture models are finite mixtures of normal distributions with an accumulation of density mass at the limits, and a gap between 100% quality of life and the next smaller utility value. The package `aldvmm` uses the likelihood and expected value functions proposed by Hernandez Alava and Wailoo (2015) using normal component distributions and a multinomial logit model of probabilities of component membership.

Usage

```
aldvmm(
  formula,
  data,
  subset = NULL,
  psi,
  ncmp = 2,
  dist = "normal",
  optim.method = NULL,
  optim.control = list(trace = FALSE),
  optim.grad = TRUE,
  init.method = "zero",
  init.est = NULL,
  init.lo = NULL,
  init.hi = NULL,
  se.fit = FALSE,
  model = TRUE,
  level = 0.95,
  na.action = "na.omit"
)
```

Arguments

formula	an object of class "formula" with a symbolic description of the model to be fitted. The model formula takes the form y ~ x1 + x2 | x1 + x4, where the | delimiter separates the model for expected values of normal distributions (left) and the multinomial logit model of probabilities of component membership (right).
data	a data frame, list or environment (or object coercible to a data frame by base::as.data.frame()) including data on outcomes and explanatory variables in 'formula'.
subset	an optional numeric vector of row indices of the subset of the model matrix used in the estimation. 'subset' can be longer than the number of rows in data and include repeated values for re-sampling purposes.
psi	a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. c(-0.594, 0.883)). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in 'psi' does not matter.
ncmp	a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.
dist	an optional character value of the distribution used in the components. In this release, only the normal distribution is available, and the default value is set to "normal".
optim.method	an optional character value of one of the following optimx::optimr() methods: "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "nlminb", "Rcgmin", "Rvmmin" and "hjn". The default method is "BFGS". The method "L-BFGS-B" is used when lower and/or upper constraints are set using 'init.lo' and 'init.hi'. The method "nlm" cannot be used in the 'aldvmm' package.
optim.control	an optional list of optimx::optimr() control parameters.
optim.grad	an optional logical value indicating if an analytical gradient should be used in optimx::optimr() methods that can use this information. The default value is TRUE. If 'optim.grad' is set to FALSE, a finite difference approximation is used.
init.method	an optional character value indicating the method for obtaining initial values. The following values are available: "zero", "random", "constant" and "sann". The default value is "zero".
init.est	an optional numeric vector of user-defined initial values. User-defined initial values override the 'init.method' argument. Initial values have to follow the same order as parameter estimates in the return value 'coef'.
init.lo	an optional numeric vector of user-defined lower limits for constrained optimization. When 'init.lo' is not NULL, the optimization method "L-BFGS-B" is used. Lower limits of parameters have to follow the same order as parameter estimates in the return value 'coef'.
init.hi	an optional numeric vector of user-defined upper limits for constrained optimization. When 'init.hi' is not NULL, the optimization method "L-BFGS-B" is used. Upper limits of parameters have to follow the same order as parameter estimates in the return value 'coef'.

se.fit	an optional logical value indicating whether standard errors of fitted values are calculated. The default value is FALSE.
model	an optional logical value indicating whether the estimation data frame is returned in the output object. The default value is TRUE.
level	a numeric value of the significance level for confidence bands of fitted values. The default value is 0.95.
na.action	a character value passed to argument 'na.action' of the function stats::model.frame() in the preparation of the model matrix. The default value is "na.omit".

Details

aldvmm() fits an adjusted limited dependent variable mixture model using the likelihood and expected value functions from Hernandez Alava and Wailoo (2015). The model accounts for latent classes, multi-modality, minimum and maximum utility values and potential gaps between 1 and the next smaller utility value. Adjusted limited dependent variable mixture models combine multiple component distributions with a multinomial logit model of the probabilities of component membership. The standard deviations of normal distributions are estimated and reported as log-transformed values which enter the likelihood function as exponentiated values to ensure non-negative values.

The minimum utility and the largest utility smaller than or equal to 1 are supplied in the argument 'psi'. The number of distributions/components that are mixed is set by the argument 'ncmp'. When 'ncmp' is set to 1 the procedure estimates a tobit model with a gap between 1 and the maximum utility value in 'psi'. The current version only allows finite mixtures of normal distributions.

The 'formula' object can include a | delimiter to separate formulae for expected values in components (left) and the multinomial logit model of probabilities of group membership (right). If no | delimiter is used, the same formula will be used for expected values in components and the multinomial logit of the probabilities of component membership.

aldvmm() uses optimx::optimr() for maximum likelihood estimation of model parameters. The argument 'optim.method' accepts the following methods: "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "nlminb", "Rcgmin", "Rvmmin" and "hjn". The default method is "BFGS". The method "nlm" cannot be used in aldvmm() because it requires a different implementation of the likelihood function. The argument 'optim.control' accepts a list of optimx::optimr() control parameters. If 'optim.grad' is set to TRUE the function optimx::optimr() uses analytical gradients during the optimization procedure for all methods that allow for this approach. If 'optim.grad' is set to FALSE or a method cannot use gradients, a finite difference approximation is used. The hessian matrix at maximum likelihood parameters is approximated numerically using numDeriv::hessian().

'init.method' accepts four values of methods for generating initial values: "zero", "random", "constant", "sann". The method "zero" sets initial values of all parameters to 0. The method "random" draws random starting values from a standard normal distribution. The method "constant" estimates a constant-only model and uses estimates as initial values of intercepts and standard errors and 0 for all other parameters. The method "sann" estimates the full model using the simulated annealing optimization method in stats::optim() and uses parameter estimates as initial values. When user-specified initial values are supplied in 'init.est', the argument 'init.method' is ignored.

By default, aldvmm() performs unconstrained optimization with upper and lower limits at -Inf and Inf. When user-defined lower and upper limits are supplied to 'init.lo' and/or 'init.hi', these default limits are replaced with the user-specified values, and the method "L-BFGS-B" is used for

box-constrained optimization instead of the user defined 'optim.method'. It is possible to only set either maximum or minimum limits. When initial values supplied to 'init.est' or from default methods lie outside the limits, the in-feasible values will be set to the limits using the function optimx::bmchk().

The function aldvmm() returns the negative log-likelihood, Akaike information criterion and Bayesian information criterion. Smaller values of these measures indicate better fit.

If 'se.fit' is set to TRUE, standard errors of fitted values are calculated using the delta method. The standard errors of fitted values in the estimation data set are calculated as $se_{fit} = \sqrt{G^t \Sigma G}$, where G is the gradient of a fitted value with respect to changes of parameter estimates, and Σ is the estimated covariance matrix of parameters (Dowd et al., 2014). The standard errors of predicted values in new data sets are calculated as $se_{pred} = \sqrt{MSE + G^t \Sigma G}$, where MSE is the mean squared error of fitted versus observed outcomes in the original estimation data (Whitmore, 1986).

The generic function base::summary() can be used to obtain or print a summary of the results. The generic function stats::predict() can be used to obtain predicted values and standard errors of predictions in new data.

Value

aldvmm() returns an object of class "aldvmm". An object of class "aldvmm" is a list containing the following objects.

coef	a numeric vector of parameter estimates.
hessian	a numeric matrix object with second partial derivatives of the likelihood function.
cov	a numeric matrix object with covariances of parameters.
n	a scalar representing the number of observations that were used in the estimation.
k	a scalar representing the number of components that were mixed.
df.null	an integer value of the residual degrees of freedom of a null model including intercepts and standard errors.
df.residual	an integer value of the residual degrees of freedom..
iter	an integer value of the number of iterations used in optimization.
convergence	an integer value indicating convergence. "0" indicates successful completion.
gof	a list including the following elements.
	ll a numeric value of the negative log-likelihood $-ll$.
	aic a numeric value of the Akaike information criterion $AIC = 2n_{par} - 2ll$.
	bic a numeric value of the Bayesian information criterion $BIC = n_{par} * log(n_{obs}) - 2ll$.
	mse a numeric value of the mean squared error $\sum (y - \hat{y})^2/(n_{obs} - n_{par})$.
	mae a numeric value of the mean absolute error $\sum \lvert y - \hat{y} \rvert/(n_{obs} - n_{par})$.
pred	a list including the following elements.
	y a numeric vector of observed outcomes in 'data'.
	yhat a numeric vector of fitted values.

`res` a numeric vector of residuals.

`se.fit` a numeric vector of the standard error of fitted values.

`lower.fit` a numeric vector of 95% lower confidence limits of fitted values.

`upper.fit` a numeric vector of 95% upper confidence limits of fitted values

`prob` a numeric matrix of expected probabilities of group membership per individual in `'data'`.

`init` a list including the following elements.

`est` a numeric vector of initial parameter estimates.

`lo` a numeric vector of lower limits of parameter estimates.

`hi` a numeric vector of upper limits of parameter estimates.

`call` a character value including the model call captured by `base::match.call()`.

`formula` an object of class "formula" supplied to argument `'formula'`.

`terms` a list of objects of class "terms" for the model of component means ("beta"), probabilities of component membership ("delta") and the full model ("full").

`contrasts` a nested list of character values showing contrasts of factors used in models of component means ("beta") and probabilities of component membership ("delta").

`data` a data frame created by `stats::model.frame` including estimation data with additional attributes.

`psi` a numeric vector with the minimum and maximum utility below 1 in `'data'`.

`dist` a character value indicating the used component distributions.

`label` a list including the following elements.

`lcoef` a character vector of labels for objects including results on distributions (default `"beta"`) and the probabilities of component membership (default `"delta"`).

`lcpar` a character vector of labels for objects including constant distribution parameters (default `"sigma"` for `dist = "normal"`).

`lcmp` a character value of the label for objects including results on different components (default "Comp")

`lvar` a list including 2 character vectors of covariate names for model parameters of distributions (`"beta"`) and the multinomial logit (`"delta"`).

`optim.method` a character value of the used `optimx::optimr()` method.

`level` a numeric value of the confidence level used for reporting.

`na.action` an object of class "omit" extracted from the "na.action" attribute of the data frame created by `stats::model.frame` in the preparation of model matrices.

References

Alava, M. H. and Wailoo, A. (2015) Fitting adjusted limited dependent variable mixture models to EQ-5D. *The Stata Journal*, **15(3)**, 737–750. doi:10.1177/1536867X1501500307

Dowd, B. E., Greene, W. H., and Norton, E. C. (2014) Computation of standard errors. *Health services research*, **49(2)**, 731–750. doi:10.1111/14756773.12122

Whitmore, G. A. (1986) Prediction limits for a univariate normal observation. *The American Statistician*, **40(2)**, 141–143. doi:10.1080/00031305.1986.10475378

Examples

```
data(utility)

fit <- aldvmm(eq5d ~ age + female | 1,
              data = utility,
              psi = c(0.883, -0.594),
              ncmp = 2)

summary(fit)

yhat <- predict(fit)
```

`aldvmm.check` *Checking the Validity of Objects Supplied to* `aldvmm()`

Description

`aldvmm.check()` runs validity checks of objects supplied to `aldvmm()`.

Usage

```
aldvmm.check(
  formula,
  data,
  subset,
  psi,
  ncmp,
  dist,
  optim.method,
  optim.control,
  optim.grad,
  init.method,
  init.est,
  init.lo,
  init.hi,
  se.fit,
  model,
  level,
  na.action,
  lcoef,
  lcpar,
  lcmp
)
```

Arguments

formula	an object of class "formula" with a symbolic description of the model to be fitted. The model formula takes the form `y ~ x1 + x2 | x1 + x4`, where the `|` delimiter separates the model for expected values of normal distributions (left) and the multinomial logit model of probabilities of component membership (right).
data	a data frame, list or environment (or object coercible to a data frame by `base::as.data.frame()`) including data on outcomes and explanatory variables in `'formula'`.
subset	an optional numeric vector of row indices of the subset of the model matrix used in the estimation. `'subset'` can be longer than the number of rows in `data` and include repeated values for re-sampling purposes.
psi	a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. `c(-0.594, 0.883)`). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in `'psi'` does not matter.
ncmp	a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in `'psi'`.
dist	an optional character value of the distribution used in the components. In this release, only the normal distribution is available, and the default value is set to `"normal"`.
optim.method	an optional character value of one of the following `optimx::optimr()` methods: `"Nelder-Mead"`, `"BFGS"`, `"CG"`, `"L-BFGS-B"`, `"nlminb"`, `"Rcgmin"`, `"Rvmmin"` and `"hjn"`. The default method is `"BFGS"`. The method `"L-BFGS-B"` is used when lower and/or upper constraints are set using `'init.lo'` and `'init.hi'`. The method `"nlm"` cannot be used in the `'aldvmm'` package.
optim.control	an optional list of `optimx::optimr()` control parameters.
optim.grad	an optional logical value indicating if an analytical gradient should be used in `optimx::optimr()` methods that can use this information. The default value is `TRUE`. If `'optim.grad'` is set to `FALSE`, a finite difference approximation is used.
init.method	an optional character value indicating the method for obtaining initial values. The following values are available: `"zero"`, `"random"`, `"constant"` and `"sann"`. The default value is `"zero"`.
init.est	an optional numeric vector of user-defined initial values. User-defined initial values override the `'init.method'` argument. Initial values have to follow the same order as parameter estimates in the return value `'coef'`.
init.lo	an optional numeric vector of user-defined lower limits for constrained optimization. When `'init.lo'` is not `NULL`, the optimization method `"L-BFGS-B"` is used. Lower limits of parameters have to follow the same order as parameter estimates in the return value `'coef'`.
init.hi	an optional numeric vector of user-defined upper limits for constrained optimization. When `'init.hi'` is not `NULL`, the optimization method `"L-BFGS-B"` is used. Upper limits of parameters have to follow the same order as parameter estimates in the return value `'coef'`.

se.fit	an optional logical value indicating whether standard errors of fitted values are calculated. The default value is FALSE.
model	an optional logical value indicating whether the estimation data frame is returned in the output object. The default value is TRUE.
level	a numeric value of the significance level for confidence bands of fitted values. The default value is 0.95.
na.action	a character value passed to argument 'na.action' of the function stats::model.frame() in the preparation of the model matrix. The default value is "na.omit".
lcoef	a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").
lcpar	a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of 'lcpar' depends on the distribution supplied to 'dist'.
lcmp	a character value representing a stub (default "Comp") for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by summary.aldvmm().

Details

aldvmm.check() checks the validity of arguments of aldvmm().

Value

aldvmm.check returns warnings or stops the execution of aldvmm() if validity checks fail.

---

aldvmm.cv *Numerical Approximation of Covariance Matrix*

Description

aldvmm.cv() performs a numerical approximation of the covariance matrix of parameter estimates.

Usage

```
aldvmm.cv(ll, par, X, y, dist, psi, ncmp, lcoef, lcpar, lcmp, optim.method)
```

Arguments

ll	a function returning the negative log-likelihood of the adjusted limited dependent variable mixture model as a scalar result (aldvmm.ll()).
par	a named numeric vector of parameter values.
X	a list of design matrices returned by aldvmm.mm(). 'X' is of length 2 and includes a design matrix for the model of component distributions and a design matrix for the model of probabilities of group membership.

y	a numeric vector of observed outcomes from complete observations in 'data' supplied to aldvmm().
dist	an optional character value of the distribution used in the components. In this release, only the normal distribution is available, and the default value is set to "normal".
psi	a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. c(-0.594, 0.883)). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in 'psi' does not matter.
ncmp	a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.
lcoef	a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").
lcpar	a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of 'lcpar' depends on the distribution supplied to 'dist'.
lcmp	a character value representing a stub (default "Comp") for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by summary.aldvmm().
optim.method	an optional character value of one of the following optimx::optimr() methods: "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "nlminb", "Rcgmin", "Rvmmin" and "hjn". The default method is "BFGS". The method "L-BFGS-B" is used when lower and/or upper constraints are set using 'init.lo' and 'init.hi'. The method "nlm" cannot be used in the 'aldvmm' package.

Details

aldvmm.cv() uses numDeriv::hessian() to calculate the hessian matrix of the log-likelihood function supplied to 'll' at parameter values supplied to 'par'.

Value

aldvmm.cv returns a list with the following objects.

hessian	a numeric matrix with second-order partial derivatives of the likelihood function 'll'.
cv	a numeric matrix with covariances/variances of parameters in 'par'.
se	a numeric vector of standard errors of parameters in 'par'.
z	a numeric vector of z-values of parameters in 'par'.
p	a numeric vector of p-values of parameter estimates.
upper	a numeric vector of upper 95% confidence limits of parameter estimates in 'par'.
lower	a numeric vector of lower 95% confidence limits of parameter estimates in 'par'.

`aldvmm.getnames` *Creating Names of Parameter Vectors*

Description

`aldvmm.getnames()` creates names of parameter vectors used in `aldvmm()`. The order of the elements in `'lcoef'` and `'lcpar'` determines the order of parameters and the structure of summary tables returned by `summary.aldvmm()`.

Usage

```
aldvmm.getnames(X, names, lcoef, lcpar, lcmp, ncmp)
```

Arguments

`X`	a list of design matrices returned by `aldvmm.mm()`. `'X'` is of length 2 and includes a design matrix for the model of component distributions and a design matrix for the model of probabilities of group membership.
`names`	a character vector of names of considered elements (distributions `"beta"`, multinomial logit `"delta"` or constant distribution parameters, i.e. `"lnsigma"`). The elements in `'names'` are combined with covariate names in `'X'` and component labels in `'lcmp'` to create a vector of names of parameter vectors.
`lcoef`	a character vector of length 2 with labels of objects including regression coefficients of component distributions (default `"beta"`) and coefficients of probabilities of component membership (default `"delta"`).
`lcpar`	a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of `'lcpar'` depends on the distribution supplied to `'dist'`.
`lcmp`	a character value representing a stub (default `"Comp"`) for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by `summary.aldvmm()`.
`ncmp`	a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in `'psi'`.

Value

a character vector of names of parameter vectors used in `aldvmm()`.

aldvmm.getpar *Extracting Parameters from Parameter Vector into Nested List.*

Description

aldvmm.getpar() extracts parameters from parameter vectors into nested lists.

Usage

```
aldvmm.getpar(par, lcoef, lcmp, lcpar, ncmp)
```

Arguments

par	a named numeric vector of parameter values.
lcoef	a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").
lcmp	a character value representing a stub (default "Comp") for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by summary.aldvmm().
lcpar	a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of 'lcpar' depends on the distribution supplied to 'dist'.
ncmp	a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.

Details

aldvmm.getpar() identifies parameters that belong to coefficients of component distributions (label "beta"), coefficients of the multinomial logit model of probabilities of component membership (label "delta"), constant parameters of component distributions (label "lnsigma" for dist=="normal"), and 1:K components (labels "Comp1" ... "CompK")) based on parameter names generated by aldvmm.getnames().

Value

a named nested list with parameter vectors for "beta", "delta" and "lnsigma" within all 1:K components. The names of the list correspond to the labels in 'lcoef', 'lcpar' and 'lcmp'.

`aldvmm.gof` *Calculating Goodness of Fit Measures*

Description

`aldvmm.gof()` calculates residual- and likelihood-based goodness of fit measures.

Usage

```
aldvmm.gof(res, par, ll)
```

Arguments

`res`	a numeric vector of residuals of all observations in the estimation data.
`par`	a named numeric vector of parameter estimates.
`ll`	a numeric value of the log-likelihood.

Details

`aldvmm.gof()` calculates mean squared errors as $MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-k}$, and mean absolute errors as $MAE = \frac{\sum_{i=1}^{n} y_i - \hat{y}_i}{n-k}$, where y_i denotes observed outcomes, $\hat{y}_i$ denotes fitted values, n denotes the sample size, and k denotes the number of parameters. The Akaike information criterion is calculated as $2k - 2ll$, and the Bayesian information criterion is calculated as $k\log(n) - 2ll$, where ll denotes the log-likelihood.

Value

`aldvmm.gof()` returns a list including the following objects.

`mse`	a numeric value of the mean squared error of observed versus fitted outcomes.
`mae`	a numeric value of the mean absolute error of observed versus fitted outcomes.
`ll`	a numeric value of the negative log-likelihood.
`aic`	a numeric value of the Akaike information criterion.
`bic`	a numeric value of the Bayesian information criterion.

aldvmm.gr *Calculating Numeric Gradients of the Negative Log-Likelihood*

Description

aldvmm.gr() calculates numerical gradients of the negative log-likelihood of the entire estimation data with respect to parameter values in 'par'.

Usage

```
aldvmm.gr(par, X, y, psi, dist, ncmp, lcoef, lcmp, lcpar, optim.method)
```

Arguments

par	a named numeric vector of parameter values.
X	a list of design matrices returned by aldvmm.mm(). 'X' is of length 2 and includes a design matrix for the model of component distributions and a design matrix for the model of probabilities of group membership.
y	a numeric vector of observed outcomes from complete observations in 'data' supplied to aldvmm().
psi	a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. c(-0.594, 0.883)). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in 'psi' does not matter.
dist	an optional character value of the distribution used in the components. In this release, only the normal distribution is available, and the default value is set to "normal".
ncmp	a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.
lcoef	a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").
lcmp	a character value representing a stub (default "Comp") for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by summary.aldvmm().
lcpar	a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of 'lcpar' depends on the distribution supplied to 'dist'.
optim.method	an optional character value of one of the following optimx::optimr() methods: "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "nlminb", "Rcgmin", "Rvmmin" and "hjn". The default method is "BFGS". The method "L-BFGS-B" is used when lower and/or upper constraints are set using 'init.lo' and 'init.hi'. The method "nlm" cannot be used in the 'aldvmm' package.

Details

`aldvmm.gr()` uses `aldvmm.sc()` to calculate analytical gradients of the negative log-likelihood.

If `'par'` includes infinite values `aldvmm.gr()` returns a gradient of zero.

Value

a named numeric vector of first derivatives of the negative log-likelihood of the data with respect to parameters in `'par'`.

`aldvmm.init` *Creating Initial Values*

Description

`aldvmm.init()` creates initial values for the minimization of the negative log-likelihood returned by `aldvmm.ll()` using `optimx::optimr()`.

Usage

```
aldvmm.init(
  X,
  y,
  psi,
  ncmp,
  dist,
  init.method,
  init.est,
  init.lo,
  init.hi,
  optim.method,
  optim.control = list(),
  optim.grad,
  lcoef,
  lcpar,
  lcmp
)
```

Arguments

`X`	a list of design matrices returned by `aldvmm.mm()`. `'X'` is of length 2 and includes a design matrix for the model of component distributions and a design matrix for the model of probabilities of group membership.
`y`	a numeric vector of observed outcomes from complete observations in `'data'` supplied to `aldvmm()`.

psi — a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. `c(-0.594, 0.883)`). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in `'psi'` does not matter.

ncmp — a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in `'psi'`.

dist — an optional character value of the distribution used in the components. In this release, only the normal distribution is available, and the default value is set to `"normal"`.

init.method — an optional character value indicating the method for obtaining initial values. The following values are available: `"zero"`, `"random"`, `"constant"` and `"sann"`. The default value is `"zero"`.

init.est — an optional numeric vector of user-defined initial values. User-defined initial values override the `'init.method'` argument. Initial values have to follow the same order as parameter estimates in the return value `'coef'`.

init.lo — an optional numeric vector of user-defined lower limits for constrained optimization. When `'init.lo'` is not `NULL`, the optimization method `"L-BFGS-B"` is used. Lower limits of parameters have to follow the same order as parameter estimates in the return value `'coef'`.

init.hi — an optional numeric vector of user-defined upper limits for constrained optimization. When `'init.hi'` is not `NULL`, the optimization method `"L-BFGS-B"` is used. Upper limits of parameters have to follow the same order as parameter estimates in the return value `'coef'`.

optim.method — an optional character value of one of the following `optimx::optimr()` methods: `"Nelder-Mead"`, `"BFGS"`, `"CG"`, `"L-BFGS-B"`, `"nlminb"`, `"Rcgmin"`, `"Rvmmin"` and `"hjn"`. The default method is `"BFGS"`. The method `"L-BFGS-B"` is used when lower and/or upper constraints are set using `'init.lo'` and `'init.hi'`. The method `"nlm"` cannot be used in the `'aldvmm'` package.

optim.control — an optional list of `optimx::optimr()` control parameters.

optim.grad — an optional logical value indicating if an analytical gradient should be used in `optimx::optimr()` methods that can use this information. The default value is `TRUE`. If `'optim.grad'` is set to `FALSE`, a finite difference approximation is used.

lcoef — a character vector of length 2 with labels of objects including regression coefficients of component distributions (default `"beta"`) and coefficients of probabilities of component membership (default `"delta"`).

lcpar — a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of `'lcpar'` depends on the distribution supplied to `'dist'`.

lcmp — a character value representing a stub (default `"Comp"`) for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by `summary.aldvmm()`.

Details

'init.method' accepts four methods for generating initial values: "zero", "random", "constant", "sann". The method "zero" sets initial values of all parameters to 0. The method "random" draws random starting values from a standard normal distribution. The method "constant" estimates a constant-only model and uses estimates as initial values for intercepts and constant distribution parameters and 0 for all other parameters. The method "sann" estimates the full model using the simulated annealing optimization method and uses all parameter estimates as initial values. When user-specified initial values are supplied in 'init.est', the argument 'init.method' is ignored.

By default, aldvmm() performs unconstrained optimization with upper and lower limits at -Inf and Inf. When user-defined lower and upper limits are supplied to 'init.lo' and/or 'init-hi', these default limits are replaced with the user-specified values, and the method "L-BFGS-B" is used for box-constrained optimization instead of the user defined 'optim.method'. It is possible to only set either maximum or minimum limits. When initial values supplied to 'init.est' or from default methods lie outside the limits, the in-feasible values will be set to the limits using the function optimx::bmchk().

Value

aldvmm.init() returns a list with the following objects.

est	a numeric vector of initial values of parameters.
lo	a numeric vector of lower limits of parameters.
hi	a numeric vector of upper limits of parameters.

---

aldvmm.ll *Calculating the Negative Log-Likelihood of the Adjusted Limited Dependent Variable Mixture Model*

Description

aldvmm.ll() calculates the negative log-likelihood of 'data' supplied to aldvmm() at the parameter values in 'par'.

Usage

```
aldvmm.ll(par, X, y, psi, ncmp, dist, lcoef, lcpar, lcmp, optim.method)
```

Arguments

par	a named numeric vector of parameter values.
X	a list of design matrices returned by aldvmm.mm(). 'X' is of length 2 and includes a design matrix for the model of component distributions and a design matrix for the model of probabilities of group membership.
y	a numeric vector of observed outcomes from complete observations in 'data' supplied to aldvmm().

psi	a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. `c(-0.594, 0.883)`). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in `'psi'` does not matter.
ncmp	a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in `'psi'`.
dist	an optional character value of the distribution used in the components. In this release, only the normal distribution is available, and the default value is set to `"normal"`.
lcoef	a character vector of length 2 with labels of objects including regression coefficients of component distributions (default `"beta"`) and coefficients of probabilities of component membership (default `"delta"`).
lcpar	a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of `'lcpar'` depends on the distribution supplied to `'dist'`.
lcmp	a character value representing a stub (default `"Comp"`) for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by `summary.aldvmm()`.
optim.method	an optional character value of one of the following `optimx::optimr()` methods: `"Nelder-Mead"`, `"BFGS"`, `"CG"`, `"L-BFGS-B"`, `"nlminb"`, `"Rcgmin"`, `"Rvmmin"` and `"hjn"`. The default method is `"BFGS"`. The method `"L-BFGS-B"` is used when lower and/or upper constraints are set using `'init.lo'` and `'init.hi'`. The method `"nlm"` cannot be used in the `'aldvmm'` package.

Details

`aldvmm.ll()` calculates the negative log-likelihood of the adjusted limited dependent variable mixture model using the likelihood function published in Hernandez Alava and Wailoo (2015). Constant distribution parameters that need to be non-negative (i.e. the standard deviations of normal distributions) enter the likelihood function as log-transformed values.

As the "L-BFGS-B" and "Rcgmin" methods in `optimx::optimr()` fail if they encounter infinite values, the log-likelihood function takes the value -1e+20 if it is infinite during these algorithms.

The names of the parameter vector supplied to `'par'` must be generated using `aldvmm.getnames()` because they will be inherited by return values of other functions in the package `'aldvmm'`. The names will also be used in the extraction of parameters from parameter vectors into nested lists using `aldvmm.getpar()`.

Value

a scalar of the negative log-likelihood of the data at parameter values in `'par'`.

References

Alava, M. H. and Wailoo, A. (2015) Fitting adjusted limited dependent variable mixture models to EQ-5D. *The Stata Journal*, **15(3)**, 737–750. doi:10.1177/1536867X1501500307

aldvmm.mm *Creating Design Matrices*

Description

aldvmm.mm() creates a list of two design matrices, one of the model of component distributions ("beta") and one of the model of probabilities of component membership ("delta").

Usage

```
aldvmm.mm(mf, Formula, ncmp, lcoef)
```

Arguments

mf	a data frame created by stats::model.frame including the variables used in formula supplied to aldvmm() plus additional attributes, including an object of class "terms" derived from 'formula'.
Formula	an object of class "Formula" created by Formula::Formula based on the formula supplied to aldvmm().
ncmp	a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.
lcoef	a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").

Details

aldvmm.mm() uses stats::model.matrix() to create design matrices for models of component distributions ("beta") and probabilities of component membership ("delta") based on a "Formula" object created by Formula::Formula and a model frame created by stats::model.frame. The design matrix for probabilities of group membership is only created if more than one components are specified in 'ncmp'.

Value

a named list of numeric matrices.

beta	a numeric design matrix for the model of component distributions.
delta	a numeric design matrix of the multinomial logit model of probabilities of component membership.

`aldvmm.pred` *Predicting Expected Values from Adjusted Limited Dependent Variable Mixture Models*

Description

`aldvmm.pred()` makes predictions of observations in design matrices in `'X'` using parameter estimates returned by `aldvmm()`.

Usage

```
aldvmm.pred(par, X, y = NULL, psi, ncmp, dist, lcoef, lcpar, lcmp)
```

Arguments

`par`	a named numeric vector of parameter values.
`X`	a list of design matrices returned by `aldvmm.mm()`. `'X'` is of length 2 and includes a design matrix for the model of component distributions and a design matrix for the model of probabilities of group membership.
`y`	a numeric vector of observed outcomes from complete observations in `'data'` supplied to `aldvmm()`.
`psi`	a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. `c(-0.594, 0.883)`). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in `'psi'` does not matter.
`ncmp`	a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in `'psi'`.
`dist`	an optional character value of the distribution used in the components. In this release, only the normal distribution is available, and the default value is set to `"normal"`.
`lcoef`	a character vector of length 2 with labels of objects including regression coefficients of component distributions (default `"beta"`) and coefficients of probabilities of component membership (default `"delta"`).
`lcpar`	a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of `'lcpar'` depends on the distribution supplied to `'dist'`.
`lcmp`	a character value representing a stub (default `"Comp"`) for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by `summary.aldvmm()`.

Details

`aldvmm.pred()` calculates expected values for observations in design matrices in `'X'` using the expected value function published in Hernandez Alava and Wailoo (2015). Constant distribution parameters that need to be non-negative (i.e. standard deviations of normal distributions) enter the expected value function as log-transformed values.

Value

a list of of predicted outcomes including the following elements.

`y`	a numeric vector of observed outcomes in `'data'`.
`yhat`	a numeric vector of fitted values.
`res`	a numeric vector of residuals.
`prob`	a numeric matrix of expected probabilities of group membership per individual in `'data'`.

`aldvmm.sc` *Calculating analytical Gradients of the Negative Log-Likelihood for each observation*

Description

`aldvmm.sc()` calculates analytical gradients of the negative log-likelihood with respect to parameter values in `'par'` for each observation in the estimation data.

Usage

```
aldvmm.sc(
  par,
  X,
  y,
  psi,
  dist,
  ncmp,
  lcoef = lcoef,
  lcmp = lcmp,
  lcpar = lcpar,
  optim.method
)
```

Arguments

`par`	a named numeric vector of parameter values.
`X`	a list of design matrices returned by `aldvmm.mm()`. `'X'` is of length 2 and includes a design matrix for the model of component distributions and a design matrix for the model of probabilities of group membership.

y	a numeric vector of observed outcomes from complete observations in 'data' supplied to aldvmm().
psi	a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. c(-0.594, 0.883)). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in 'psi' does not matter.
dist	an optional character value of the distribution used in the components. In this release, only the normal distribution is available, and the default value is set to "normal".
ncmp	a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.
lcoef	a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").
lcmp	a character value representing a stub (default "Comp") for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by summary.aldvmm().
lcpar	a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of 'lcpar' depends on the distribution supplied to 'dist'.
optim.method	an optional character value of one of the following optimx::optimr() methods: "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "nlminb", "Rcgmin", "Rvmmin" and "hjn". The default method is "BFGS". The method "L-BFGS-B" is used when lower and/or upper constraints are set using 'init.lo' and 'init.hi'. The method "nlm" cannot be used in the 'aldvmm' package.

Details

aldvmm.sc() calculates gradients of the negative log-likelihood.

Value

a named numeric matrix of first derivatives of the negative log-likelihood of the data with respect to parameters in 'par'.

---

aldvmm.sefit *Calculating Standard Errors of Fitted and Predicted Outcomes*

Description

aldvmm.sefit() calculates standard errors of fitted and predicted outcomes using the delta method.

Usage

```
aldvmm.sefit(
  par,
  yhat,
  X,
  type,
  psi,
  cv,
  mse = NA,
  ncmp,
  dist,
  level,
  lcoef,
  lcmp,
  lcpar
)
```

Arguments

par	a named numeric vector of parameter values.
yhat	a numeric vector of predicted outcomes returned by `aldvmm.pred()`.
X	a list of design matrices returned by `aldvmm.mm()`. `'X'` is of length 2 and includes a design matrix for the model of component distributions and a design matrix for the model of probabilities of group membership.
type	a character value of either 'fit' or 'pred' indicating whether the standard error of the fit ('fit') or the standard error of predictions in new data ('pred') are calculated.
psi	a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. `c(-0.594, 0.883)`). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in `'psi'` does not matter.
cv	a numeric matrix with covariances/variances of parameter estimates returned by `aldvmm.cv()`.
mse	a numeric value of the mean squared error of observed versus predicted outcomes $\sum (y - \hat{y})^2/(n_{obs} - n_{par})$ for all observations in model matrices `'X'` supplied to `aldvmm.ll()`.
ncmp	a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in `'psi'`.
dist	an optional character value of the distribution used in the components. In this release, only the normal distribution is available, and the default value is set to `"normal"`.
level	a numeric value of the significance level for confidence bands of fitted values. The default value is 0.95.

`lcoef` a character vector of length 2 with labels of objects including regression coefficients of component distributions (default `"beta"`) and coefficients of probabilities of component membership (default `"delta"`).

`lcmp` a character value representing a stub (default `"Comp"`) for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by `summary.aldvmm()`.

`lcpar` a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of `'lcpar'` depends on the distribution supplied to `'dist'`.

Details

`aldvmm.sefit()` calculates standard errors of fitted values using the delta method. Standard errors of fitted values in the estimation data set are calculated as $se_{fit} = \sqrt{G^t \Sigma G}$, where G is the gradient of a fitted value with respect to changes of parameter estimates, and Σ is the estimated covariance matrix of parameters (Dowd et al., 2014). Standard errors of predicted values in new data sets are calculated as $se_{pred} = \sqrt{MSE + G^t \Sigma G}$, where MSE is the mean squared error of fitted versus observed outcomes in the original estimation data (Whitmore, 1986). The gradients of fitted values with respect to parameter estimates are approximated numerically using `numDeriv::jacobian()`.

Value

a named numeric vector of standard errors of fitted or predicted outcomes. The names of the elements in the vector are identical to the row names of design matrices in `'X'`

References

Whitmore, G. A. (1986). Prediction limits for a univariate normal observation. The American Statistician, 40(2), 141-143. https://doi.org/10.1080/00031305.1986.10475378

Dowd, B. E., Greene, W. H., and Norton, E. C. (2014) Computation of standard errors. *Health services research*, **49(2)**, 731–750. doi:10.1111/14756773.12122

`aldvmm.sum` *Creating Summary Table*

Description

`aldvmm.sum()` creates a summary table of regression results.

Usage

```
aldvmm.sum(object, digits = max(3L, getOption("digits") - 3L), level = 0.95)
```

Arguments

object	an aldvmm() model fit object of class "aldvmm".
digits	a numeric value of the number of digits in the reporting table.
level	a numeric value of the confidence level.

Value

a data.frame object with a summary table of regression results.

---

aldvmm.tm *Creating Terms Objects*

Description

aldvmm.tm() creates a list of up to three "terms" objects, one of the model of component distributions ("beta"), one of the model of probabilities of component membership ("delta") and one for the full model ("full").

Usage

```
aldvmm.tm(mf, Formula, ncmp, lcoef)
```

Arguments

mf	a data frame created by stats::model.frame including the variables used in 'formula' supplied to aldvmm() plus additional attributes derived from 'formula'.
Formula	an object of class "Formula" created by Formula::Formula based on the 'formula' supplied to aldvmm().
ncmp	a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.
lcoef	a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").

Details

aldvmm.tm() uses stats::terms() to create "terms" objects based on a "Formula" object created by Formula::Formula and a model frame created by stats::model.frame. The "terms" object for probabilities of group membership is only created if more than one components are specified in 'ncmp'.

Value

a named list of objects of class "terms".

`beta`	a "terms" object for the model of component distributions.
`delta`	a "terms" object of the multinomial logit model of probabilities of component membership.
`full`	a "terms" object of the full model.

`coef.aldvmm` *Extract Coefficients of Adjusted Limited Dependent Variable Mixture Model Fits*

Description

The method `coef.aldvmm` for the generic function `stats::coef()` extracts the vector of coefficients from an object of class "aldvmm".

Usage

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

Arguments

`object`	an object inheriting from class "aldvmm".
`...`	further arguments passed to or from other methods.

Value

a named numeric vector of parameter estimates.

`estfun.aldvmm` *Create Matrix of Adjusted Limited Dependent Variable Mixture Model Gradients per Observation*

Description

The method `estfun.aldvmm` for the generic function `sandwich::estfun()` calculates the gradient of the aldvmm log-likelihood using with respect to parameter values for each observation `aldvmm::aldvmm.sc()`.

Usage

```
## S3 method for class 'aldvmm'
estfun(x, ...)
```

Arguments

x	an object inheriting from class "aldvmm".
...	further arguments passed to or from other methods.

Value

a numeric matrix of gradients with one row per observation and one column per parameter.

---

formula.aldvmm *Extract Adjusted Limited Dependent Variable Mixture Model Formula*

Description

The method `formula.aldvmm` for the generic function `stats::formula()` returns the formula object from an object of class "aldvmm".

Usage

```
## S3 method for class 'aldvmm'
formula(x, ...)
```

Arguments

x	an object inheriting from class "aldvmm".
...	further arguments passed to or from other methods.

Value

an object of class "formula"

---

model.matrix.aldvmm *Extract Adjusted Limited Dependent Variable Mixture Model Model Matrices*

Description

The method `model.matrix.aldvmm` for the generic function `stats::model.matrix()` extracts a list of model matrices from an object of class "aldvmm" using the function `aldvmm::aldvmm.mm()`.

Usage

```
## S3 method for class 'aldvmm'
model.matrix(object, ...)
```

Arguments

`object`	an object inheriting from class "aldvmm".
`...`	further arguments passed to or from other methods.

Value

a named list of numeric matrices.

`beta`	a numeric design matrix for the model of component distributions.
`delta`	a numeric design matrix of the multinomial logit model of probabilities of component membership.

`nobs.aldvmm` *Extract Adjusted Limited Dependent Variable Mixture Model Number of Observations*

Description

The method `nobs.aldvmm` for the generic function `stats::nobs()` extracts the number of observations from an object of class "aldvmm".

Usage

```
## S3 method for class 'aldvmm'
nobs(object, ...)
```

Arguments

`object`	an object inheriting from class "aldvmm".
`...`	further arguments passed to or from other methods.

Value

a scalar of the number of complete observations.

`predict.aldvmm` *Predict Method for Adjusted Limited Dependent Variable Mixture Model Fits*

Description

The method `predict.aldvmm` for the generic function `stats::predict()` calls `aldvmm.pred()` and `aldvmm.sefit()` to predict outcomes and their standard errors in new data.

Usage

```
## S3 method for class 'aldvmm'
predict(
  object,
  newdata = NULL,
  se.fit = FALSE,
  type = "pred",
  level = 0.95,
  ...
)
```

Arguments

`object`	an object inheriting from class 'aldvmm'.
`newdata`	a data frame, list or environment (or object coercible to a data frame by `base::as.data.frame()`) including explanatory variables for prediction.
`se.fit`	an optional logical value indicating whether standard errors of fitted values are calculated. The default value is `FALSE`.
`type`	a character value of either 'fit' or 'pred' indicating whether the standard error of the fit ('fit') or the standard error of predictions in new data ('pred') are calculated.
`level`	a numeric value of the significance level for confidence bands of fitted values. The default value is 0.95.
`...`	further arguments passed to or from other methods.

Value

a named list of numeric vectors of predicted outcomes, standard errors and confidence or prediction intervals.

print.aldvmm *Print Adjusted Limited Dependent Variable Mixture Model Fits*

Description

The method `print.aldvmm` for the generic function `base::print()` prints a summary of an object of class "aldvmm".

Usage

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

Arguments

`x`	an object inheriting from class "aldvmm".
`digits`	an integer value of the number of digits in the printed output.
`...`	further arguments passed to or from other methods.

print.summary.aldvmm *Printing Adjusted Limited Dependent Variable Mixture Model Summaries*

Description

The method `print.summary.aldvmm` for the generic function `base::print()` prints a summary of an object of class "summary.aldvmm".

Usage

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

Arguments

`x`	an object inheriting from class 'summary.aldvmm'.
`digits`	an integer value of the number of digits in the output table.
`...`	further arguments passed to or from other methods.

residuals.aldvmm — *Extract Adjusted Limited Dependent Variable Mixture Model Residuals*

Description

The method `residuals.aldvmm` for the generic function `stats::residuals()` returns the covariance matrix from an object of class "aldvmm".

Usage

```
## S3 method for class 'aldvmm'
residuals(object, ...)
```

Arguments

`object`	an object inheriting from class "aldvmm".
`...`	further arguments passed to or from other methods.

Value

a numeric vector of residuals.

summary.aldvmm — *Summarizing Adjusted Limited Dependent Variable Mixture Model Fits*

Description

The method `summary.aldvmm` for the generic function `base::summary()` creates an object of class "summary.aldvmm" including key results from an object of class "aldvmm".

Usage

```
## S3 method for class 'aldvmm'
summary(object, digits = max(3L, getOption("digits") - 3L), level = 0.95, ...)
```

Arguments

`object`	an object inheriting from class 'aldvmm'.
`digits`	an integer value of the number of digits in the output table.
`level`	a numeric value of the confidence interval between 0 and 1.
`...`	further arguments passed to or from other methods.

Value

summary.aldvmm returns an object of class "summary.aldvmm" including the following elements.

call	a character value including the model call captured by base::match.call.
summary	a data frame generated by aldvmm::aldvmm.sum including a formatted summary of model results for printing.
terms	a list of objects of class stats::terms.
contrasts	a nested list of character values showing contrasts of factors used in models of component means ("beta") and probabilities of component membership ("delta").
coef	a numeric vector of parameter estimates.
cov	a numeric matrix object with covariances of parameters.
n	a scalar representing the number of complete observations with no missing values that were used in the estimation.
df.residual	an integer value of the residual degrees of freedom.
df.null	an integer value of the residual degrees of freedom of a null model including intercepts and standard errors.
iter	an integer value of the number of iterations used in optimization.
ll	a numeric value of the negative log-likelihood $-ll$.
aic	a numeric value of the Akaike information criterion $AIC = 2n_{par} - 2ll$.
bic	a numeric value of the Bayesian information criterion $BIC = n_{par} * log(n_{obs}) - 2ll$.
k	a numeric value of the number of components.
lcoef	a character vector of labels for objects including results on distributions (default "beta") and the probabilities of component membership (default "delta").
lcpar	a character vector of labels for objects including constant distribution parameters (default "sigma" for dist = "normal").
lcmp	a character value of the label for objects including results on different components (default "Comp")
lvar	a list including 2 character vectors of covariate names for model parameters of distributions ("beta") and the multinomial logit ("delta").

---

terms.aldvmm *Extract Adjusted Limited Dependent Variable Mixture Model Terms*

Description

The method terms.aldvmm for the generic function stats::terms() returns the terms object for the combined model of component means and probabilities of component membership from an object of class "aldvmm".

Usage

```
## S3 method for class 'aldvmm'
terms(x, ...)
```

Arguments

x	an object inheriting from class "aldvmm".
...	further arguments passed to or from other methods.

Value

an object of class "terms"

`update.aldvmm` *Update Adjusted Limited Dependent Variable Mixture Model Fit*

Description

The method `update.aldvmm` for the generic function `stats::update()` re-estimates an object of class "aldvmm".

Usage

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

Arguments

object	an object inheriting from class "aldvmm".
formula.	a formula object representing the new model.
...	further arguments passed to or from other methods.
evaluate	a logical value indicating if the model should be re-estimated (TRUE) or not (FALSE).

Value

an object of class "aldvmm"

`utility` *Simulated Example Data of Health State Utilities.*

Description

`utility` is a simulated data frame including health state utilities and patients' age and sex.

Usage

```
utility
```

Format

A data frame with 200 rows and 3 variables:

`utility`	a utility value {[-0.594, 0.883], 1}.
`age`	Age in years.
`female`	Indicator of female sex.

Examples

```
set.seed(101010101)
utility                <- data.frame(female = rbinom(size = 1,
                                     n      = 200,
                                     p      = 0.6))
utility[, 'age']       <- stats::rnorm(n    = 200,
                                       mean = 50 + utility$female*10,
                                       sd   = 15)
utility[1:50, 'eq5d']  <- stats::rnorm(n    = 50,
                                       mean = 0 - 0.1 *
                                              utility[1:50, 'female'] +
                                              0.0005 * utility[1:50, 'age'],
                                       sd   = 0.1)
utility[51:200, 'eq5d'] <- stats::rnorm(n    = 150,
                                       mean = 0.5 +
                                          0.1 * utility[51:200, 'female'] +
                                          0.0001*utility[51:200, 'age'],
                                       sd   = 0.2)
utility[utility$eq5d<(-0.594), 'eq5d'] <- -0.594
utility[utility$eq5d>0.883, 'eq5d'] <- 1
hist(utility$eq5d, breaks = 50)
```

`vcov.aldvmm` *Extract Adjusted Limited Dependent Variable Mixture Model Covariance Matrix*

Description

The method `vcov.aldvmm` for the generic function `stats::vcov()` returns the covariance matrix from an object of class "aldvmm".

Usage

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

Arguments

`object`	an object inheriting from class "aldvmm".
`...`	further arguments passed to or from other methods.

Value

a numeric matrix.

Index