Package 'aldvmm'

May 6, 2021

Type Package

Title Adjusted Limited Dependent Variable Mixture Models

Version 0.8.2

Date 2021-05-06

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 + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports numDeriv, stats, checkmate, optimr

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

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

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

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Mark Pletscher [aut, cre, cph]

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

R topics documented:

1

---

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

summary(fit)

 yhat <- predict(fit,
                 newdata = utility)
```

---

aldvmm   *Fitting Adjusted Limited Dependent Variable Mixture Models*

Description

The function `aldvmm::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,
  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,
  level = 0.95
)
```

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) (see details).
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'`.
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 finite mixture. In this release, only the normal distribution is available, and the default value is ″normal″.
optim.method	an optional character value of one of the following optimr::optimr() methods: ″Nelder-Mead″, ″BFGS″, ″CG″, ″L-BFGS-B″, ″nlminb″, ″Rcgmin″, ″Rvmmin″ and ″hjn″. The default method is ″Nelder-Mead″. 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 (see details).
optim.control	an optional list of optimr::optimr() control parameters.
optim.grad	an optional logical value indicating if a numeric gradient should be used in optimr::optimr() methods that can use this information. The default value is TRUE. If 'optim.grad' is set to FALSE, a finite difference approximation will be used.
init.method	an optional character value indicating the method for obtaining initial values (see details). 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 'par'.
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 'par'.
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 'par'.
se.fit	an optional logical value indicating whether standard errors of the fit are calculated (see Details). The default value is FALSE.
level	a numeric value of the significance level for confidence bands of fitted values. The default value is 0.95.

Details

aldvmm::aldvmm() fits adjusted limited dependent variable mixture models using the likelihood and expected value functions from Hernandez Alava and Wailoo (2015). The model accounts for multi-modality, minimum and maximum utility values and potential gaps between 1 and the next smaller observed utility value. Adjusted limited dependent variable mixture models combine multiple distributions in different components 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::aldvmm() uses optimr::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 "Nelder-Mead". The method "nlm" cannot be used in aldvmm::aldvmm() because it requires a different implementation of the likelihood function. The argument 'optim.control' accepts a list of optimr::optimr() control parameters. If 'optim.grad' is set to TRUE the function optimr::optimr() uses numerical 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 numerical gradients of the likelihood function are approximated numerically using the function numDeriv::grad(). 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 prameters. 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::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.

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 using the method

aldvmm::predict.aldvmm() 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

aldvmm::aldvmm() returns an object of class inheriting from "aldvmm".

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.

An object of class "aldvmm" is a list containing at least the following objects.

coef	a numeric vector of parameter estimates.
se	a numeric vector of standard errors of parameter estimates.
z	a numeric vector of standardized parameter estimates.
p	a numeric vector of p-values of parameter estimates.
lower	a numeric vector of 95% lower confidence limits of parameter estimates.

upper	a numeric vector of 95% upper confidence limits 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 complete observations with no missing values that were used in the estimation.
k	a scalar representing the number of components that were mixed.
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 $\sum |y - \hat{y}| / (n_{obs} - n_{par})$.

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 vector expected values of the probabilities of group membership.

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.

formula	an object of class stats::formula supplied to argument 'formula'.
psi	a numeric vector with the minimum and maximum utility below 1 in 'data'.
dist	a character value indicating the used distribution.
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 optimr::optimr() method.

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,
                newdata = utility)
```

---

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

Description

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

Usage

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

```
    level
)
```

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) (see details).
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'.
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 finite mixture. In this release, only the normal distribution is available, and the default value is ″normal″.
lcoef	a character vector of length 2 with labels of objects including regression coefficients. The first element is the label of coefficients for component distributions (default ″beta″), and the second element is the label of coefficients on 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 aldvmm::summary.aldvmm().
init.method	an optional character value indicating the method for obtaining initial values (see details). The following values are available: ″zero″, ″random″, ″constant″ and ″sann″. The default value is ″zero″.
optim.method	an optional character value of one of the following optimr::optimr() methods: ″Nelder-Mead″, ″BFGS″, ″CG″, ″L-BFGS-B″, ″nlminb″, ″Rcgmin″, ″Rvmmin″ and ″hjn″. The default method is ″Nelder-Mead″. 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 (see details).
optim.grad	an optional logical value indicating if a numeric gradient should be used in optimr::optimr() methods that can use this information. The default value is TRUE. If 'optim.grad' is set to FALSE, a finite difference approximation will be used.
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 'par'.

init.lo   an optional numeric vector of user-defined lower limits for constrained opti-
             mization. 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 `'par'`.

init.hi   an optional numeric vector of user-defined upper limits for constrained opti-
             mization. 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 `'par'`.

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

se.fit   an optional logical value indicating whether standard errors of the fit are calcu-
             lated (see Details). The default value is `FALSE`.

level   a numeric value of the significance level for confidence bands of fitted values.
             The default value is 0.95.

Details

`aldvmm::aldvmm.check()` checks the validity of input values of `aldvmm::aldvmm()` and of user
settings.

Value

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

---

aldvmm.cv   *Numerical Approximation of Covariance Matrix*

Description

`aldvmm::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 `aldvmm::aldvmm.ll()` returning the negative log-likelihood of the
             adjusted limited dependent variable mixture model as a scalar result.

par   a named numeric vector of parameter values used to calculate the log-likelihood.
             The names of the parameter vector must be generated using

             `aldvmm::aldvmm.getnames()` because they will be inherited by return values
             of functions in package `'aldvmm'`, and the names will be used in the extraction
             of parameters into nested lists using `aldvmm::aldvmm.getpar()`.

X   a list of design matrices returned by `aldvmm::aldvmm.mm()`. `'X'` is of length
             2 and includes numeric design matrices for the models of component distribu-
             tions (label `"beta"`) and the probabilities of group membership (label `"delta"`).
             The numeric matrices in `'X'` have rownames inherited from `'data'` supplied to
             `aldvmm::aldvmm()`. The rownames will be used in

	aldvmm::aldvmm.pred() to return missing predictions for incomplete observations in 'data'.
y	a numeric vector of observed outcomes from complete observations in 'data' supplied to aldvmm::aldvmm().
dist	an optional character value of the distribution used in the finite mixture. In this release, only the normal distribution is available, and the default value is "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. The first element is the label of coefficients for component distributions (default "beta"), and the second element is the label of coefficients on 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 aldvmm::summary.aldvmm().
optim.method	an optional character value of one of the following optimr::optimr() methods: "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "nlminb", "Rcgmin", "Rvmmin" and "hjn". The default method is "Nelder-Mead". 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 (see details).

Details

aldvmm::aldvmm.cv() uses numDeriv::hessian() to calculate the hessian matrix of the log-likelihood function in 'll' at parameter values in '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 (parameter values in 'par' divided by standard error 'se').
p	a numeric vector of p-values of parameter estimates (probabilities of 'z' smaller than zero).

upper	a numeric vector of upper 95% confindence limits of paramter estimates in `'par'`.
lower	a numeric vector of lower 95% confindence limits of paramter estimates in `'par'`.

---

aldvmm.getnames	*Creating Names of Parameter Vectors*

Description

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

Usage

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

Arguments

X	a list of design matrices returned by `aldvmm::aldvmm.mm()`. `'X'` is of length 2 and includes numeric design matrices for the models of component distributions (label ″`beta`″) and the probabilities of group membership (label ″`delta`″). The numeric matrices in `'X'` have rownames inherited from `'data'` supplied to `aldvmm::aldvmm()`. The rownames will be used in `aldvmm::aldvmm.pred()` to return missing predictions for incomplete observations in `'data'`.
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. The first element is the label of coefficients for component distributions (default ″`beta`″), and the second element is the label of coefficients on 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 `aldvmm::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()`.

---

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 used to calculate the log-likelihood.
           The names of the parameter vector must be generated using

           `aldvmm::aldvmm.getnames()` because they will be inherited by return values
           of functions in package `'aldvmm'`, and the names will be used in the extraction
           of parameters into nested lists using `aldvmm::aldvmm.getpar()`.

lcoef   a character vector of length 2 with labels of objects including regression coeffi-
           cients. The first element is the label of coefficients for component distributions
           (default ″beta″), and the second element is the label of coefficients on proba-
           bilities of component membership (default ″delta″).

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

           `aldvmm::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 maxi-
           mum value in `'psi'`.

Details

`aldvmm::aldvmm.getpar()` identifies parameters that belong to coefficients of component distri-
butions (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::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.gr | *Calculating Numeric Gradients of the Negative Log-Likelihood*

Description

`aldvmm::aldvmm.gr()` calculates numeric gradients of the negative log-likelihood returned by `aldvmm::aldvmm.ll()` 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 used to calculate the log-likelihood. The names of the parameter vector must be generated using

  `aldvmm::aldvmm.getnames()` because they will be inherited by return values of functions in package `'aldvmm'`, and the names will be used in the extraction of parameters into nested lists using `aldvmm::aldvmm.getpar()`.

X
: a list of design matrices returned by `aldvmm::aldvmm.mm()`. `'X'` is of length 2 and includes numeric design matrices for the models of component distributions (label ″`beta`″) and the probabilities of group membership (label ″`delta`″). The numeric matrices in `'X'` have rownames inherited from `'data'` supplied to `aldvmm::aldvmm()`. The rownames will be used in

  `aldvmm::aldvmm.pred()` to return missing predictions for incomplete observations in `'data'`.

y
: a numeric vector of observed outcomes from complete observations in `'data'` supplied to `aldvmm::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 finite mixture. In this release, only the normal distribution is available, and the default value is ″`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. The first element is the label of coefficients for component distributions (default ″`beta`″), and the second element is the label of coefficients on 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

  `aldvmm::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 `optimr::optimr()` methods:
                  ″Nelder-Mead″, ″BFGS″, ″CG″, ″L-BFGS-B″, ″nlminb″, ″Rcgmin″, ″Rvmmin″
                  and ″hjn″. The default method is ″Nelder-Mead″. 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 (see details).

Details

`aldvmm::aldvmm.gr()` uses `numDeriv::grad()` to perform numerical approximation of gradients
of the negative log-likelihood returned by `aldvmm::aldvmm.ll()`.

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

Usage

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

Arguments

X	a list of design matrices returned by aldvmm::aldvmm.mm(). 'X' is of length 2 and includes numeric design matrices for the models of component distributions (label "beta") and the probabilities of group membership (label "delta"). The numeric matrices in 'X' have rownames inherited from 'data' supplied to aldvmm::aldvmm(). The rownames will be used in aldvmm::aldvmm.pred() to return missing predictions for incomplete observations in 'data'.
y	a numeric vector of observed outcomes from complete observations in 'data' supplied to aldvmm::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 finite mixture. In this release, only the normal distribution is available, and the default value is "normal".
init.method	an optional character value indicating the method for obtaining initial values (see details). 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 'par'.
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 'par'.
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 'par'.
optim.method	an optional character value of one of the following optimr::optimr() methods: "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "nlminb", "Rcgmin", "Rvmmin" and "hjn". The default method is "Nelder-Mead". 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 (see details).
optim.control	an optional list of optimr::optimr() control parameters.
optim.grad	an optional logical value indicating if a numeric gradient should be used in optimr::optimr() methods that can use this information. The default value is TRUE. If 'optim.grad' is set to FALSE, a finite difference approximation will be used.
lcoef	a character vector of length 2 with labels of objects including regression coefficients. The first element is the label of coefficients for component distributions (default "beta"), and the second element is the label of coefficients on 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 in-
                  cluding regression coefficients in different components (e.g. "Comp1", "Comp2",
                  ...). This label is also used in summary tables returned by

                  `aldvmm::summary.aldvmm()`.

Details

`'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 an-
nealing 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::aldvmm()` performs unconstrained optimization with upper and lower lim-
its 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.

Value

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

est   a numeric vector of initial values of parameters supplied to `optimr::optimr()`
                  in `aldvmm::aldvmm()`.

lo   a numeric vector of lower limits of parameters in `optimr::optimr()`.

hi   a numeric vector of upper limits of parameters in `optimr::optimr()`.

---

aldvmm.ll   *Calculating the Negative Adjusted Limited Dependent Variable Mix-*
                               ture Model Log-Likelihood

Description

`aldvmm::aldvmm.ll()` calculates the negative log-likelihood of `'data'` supplied to `aldvmm::aldvmm()`
for 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 used to calculate the log-likelihood. The names of the parameter vector must be generated using
	`aldvmm::aldvmm.getnames()` because they will be inherited by return values of functions in package `'aldvmm'`, and the names will be used in the extraction of parameters into nested lists using `aldvmm::aldvmm.getpar()`.
X	a list of design matrices returned by `aldvmm::aldvmm.mm()`. `'X'` is of length 2 and includes numeric design matrices for the models of component distributions (label `"beta"`) and the probabilities of group membership (label `"delta"`). The numeric matrices in `'X'` have rownames inherited from `'data'` supplied to `aldvmm::aldvmm()`. The rownames will be used in
	`aldvmm::aldvmm.pred()` to return missing predictions for incomplete observations in `'data'`.
y	a numeric vector of observed outcomes from complete observations in `'data'` supplied to `aldvmm::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 finite mixture. In this release, only the normal distribution is available, and the default value is `"normal"`.
lcoef	a character vector of length 2 with labels of objects including regression coefficients. The first element is the label of coefficients for component distributions (default `"beta"`), and the second element is the label of coefficients on 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
	`aldvmm::summary.aldvmm()`.
optim.method	an optional character value of one of the following `optimr::optimr()` methods: `"Nelder-Mead"`, `"BFGS"`, `"CG"`, `"L-BFGS-B"`, `"nlminb"`, `"Rcgmin"`, `"Rvmmin"` and `"hjn"`. The default method is `"Nelder-Mead"`. 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 (see details).

Details

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

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

Value

a scalar of the negative log-likelihood of the data for the 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::aldvmm.mm()` creates design matrices for regressions of component distributions (`"beta"`) and probabilities of component membership (`"delta"`).

Usage

```
aldvmm.mm(data, formula, ncmp, lcoef)
```

Arguments

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'`.
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) (see details).
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. The first element is the label of coefficients for component distributions (default `"beta"`), and the second element is the label of coefficients on probabilities of component membership (default `"delta"`).

Details

`aldvmm::aldvmm.mm()` uses `stats::model.matrix()` to create design matrices for regressions of component distributions (`"beta"`) and probabilities of component membership (`"delta"`) based on `'formula'` supplied to `aldvmm::aldvmm.ll()`. The design matrix for probabilities of group membership is only created if more than one component are specified in `'ncmp'`.

Value

a named list of numeric matrices.

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

---

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

Description

`aldvmm::aldvmm.pred()` makes predictions of observations in design matrices in `'X'` using the results of `aldvmm::aldvmm()`.

Usage

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

Arguments

par	a named numeric vector of parameter values used to calculate the log-likelihood. The names of the parameter vector must be generated using
	`aldvmm::aldvmm.getnames()` because they will be inherited by return values of functions in package `'aldvmm'`, and the names will be used in the extraction of parameters into nested lists using `aldvmm::aldvmm.getpar()`.
X	a list of design matrices returned by `aldvmm::aldvmm.mm()`. `'X'` is of length 2 and includes numeric design matrices for the models of component distributions (label `"beta"`) and the probabilities of group membership (label `"delta"`). The numeric matrices in `'X'` have rownames inherited from `'data'` supplied to `aldvmm::aldvmm()`. The rownames will be used in
	`aldvmm::aldvmm.pred()` to return missing predictions for incomplete observations in `'data'`.
y	a numeric vector of observed outcomes from complete observations in `'data'` supplied to `aldvmm::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 finite mixture. In this release, only the normal distribution is available, and the default value is `"normal"`.

lcoef   a character vector of length 2 with labels of objects including regression coeffi-
                     cients. The first element is the label of coefficients for component distributions
                     (default ″beta″), and the second element is the label of coefficients on proba-
                     bilities 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 in-
                     cluding regression coefficients in different components (e.g. "Comp1", "Comp2",
                     ...). This label is also used in summary tables returned by
                     aldvmm::summary.aldvmm().

Details

aldvmm::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 dis-
tribution 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 named numeric vector of predicted outcomes. The names of the elements in the vector are identical
to the row names of design matrices in 'X'.

---

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

Description

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

Usage

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

Arguments

par	a named numeric vector of parameter values used to calculate the log-likelihood. The names of the parameter vector must be generated using
	aldvmm::aldvmm.getnames() because they will be inherited by return values of functions in package 'aldvmm', and the names will be used in the extraction of parameters into nested lists using aldvmm::aldvmm.getpar().
yhat	a numeric vector of predicted outcomes from aldvmm::aldvmm.pred().
X	a list of design matrices returned by aldvmm::aldvmm.mm(). 'X' is of length 2 and includes numeric design matrices for the models of component distributions (label "beta") and the probabilities of group membership (label "delta"). The numeric matrices in 'X' have rownames inherited from 'data' supplied to aldvmm::aldvmm(). The rownames will be used in
	aldvmm::aldvmm.pred() to return missing predictions for incomplete observations in 'data'.
type	a character value from 'fit' or 'pred' indicating whether the standard error of the fit ('fit') or the standard error of predictions in new data ('pred') are calculated.
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) (see details).
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::aldvmm.cv().
mse	a scalar 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 data supplied to aldvmm::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 finite mixture. In this release, only the normal distribution is available, and the default value is "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. The first element is the label of coefficients for component distributions (default "beta"), and the second element is the label of coefficients on 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
	aldvmm::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::aldvmm.sefit()` calculates standard errors of fitted values 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 using the method

`aldvmm::predict.aldvmm()` 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 numeri-
cally using `numDeriv::jacobian()`.

Value

a named numeric vector of standard errors of fitted or predicted outcomes. The names of the ele-
ments 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::aldvmm.sum()` creates a summary table of regression results.

Usage

```
aldvmm.sum(
  est,
  se,
  z,
  p,
  lower,
  upper,
  n,
  value,
  aic,
  bic,
  ncmp,
  lcoef,
```

```
    lcpar,
    lcmp,
    lvar,
    digits = 3
)
```

Arguments

est	a named numeric vector of parameters with names returned by `aldvmm::aldvmm.getnames()`.
se	a named numeric vector of standard errors of parameters returned by `aldvmm::aldvmm.cv()`.
z	a named numeric vector of z scores (standardized coefficients) of parameters returned by `aldvmm::aldvmm.cv()`.
p	a named numeric vector of p-values of parameters returned by `aldvmm::aldvmm.cv()`.
lower	a named numeric vector of 95% lower limits of parameters returned by `aldvmm::aldvmm.cv()`.
upper	a named numeric vector of 95% upper limits of parameters returned by `aldvmm::aldvmm.cv()`.
n	a numeric value of the number of complete observations in `'data'` supplied to `aldvmm::aldvmm()`.
value	a numeric value of the negative log-likelihood returned by `aldvmm::aldvmm.ll()`.
aic	a numeric value of the Akaike information criterion (AIC) returned by `aldvmm::aldvmm()`.
bic	a numeric value of the Bayesian information criterion (BIC) returned by `aldvmm::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. The first element is the label of coefficients for component distributions (default `"beta"`), and the second element is the label of coefficients on 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 `aldvmm::summary.aldvmm()`.
lvar	a named list of character vectors with column names of design matrices returned by `aldvmm::aldvmm.mm()`.
digits	a numeric value of the number of digits in the reporting table.

Value

a data frame with a summary table of regression results.

---

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

Description

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

Usage

```
## S3 method for class 'aldvmm'
predict(object, newdata, 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 the fit are calculated (see Details). The default value is `FALSE`.
type	a character value from '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.

---

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

Description

The generic function `base::summary()` calls `aldvmm::aldvmm.sum()` to print the summary table data frame returned by `aldvmm::aldvmm.sum()` of the results.

Usage

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

Arguments

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

---

utility   *Simulated Example Data of Health State Utilities.*

Description

aldvmm::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)
```

Index