Package: xreg2 (via r-universe)

June 3, 2026

Type Package

Title Flexible Maximum Likelihood Regression with Gradient-Based Optimisation

Version 0.1.4

Description Fits flexible maximum likelihood regression models supporting censored, interval, and hybrid continuous/dichotomous data. Provides explicit analytic and numerical gradient computation, random intercept models via Gauss-Hermite quadrature, and multiple distribution families.

URL https://github.com/MathsinHealth/xreg2

Depends R (>= 4.0.0)

Imports ucminf, numDeriv

Suggests optimx, testthat (>= 3.0.0)

License GPL-2

Encoding UTF-8

RoxygenNote 7.3.3

Config/testthat/edition 3

NeedsCompilation no

Author Kim Rand [aut, cre]

Maintainer Kim Rand <krand@krand.no>

Repository https://mathsinhealth.r-universe.dev

Date/Publication 2026-04-23 12:56:01 UTC

RemoteUrl https://github.com/mathsinhealth/xreg2

RemoteRef HEAD

RemoteSha 3496e82b032d5ae3f1603be4c9b3bfba98b11093

Contents

`AIC.xreg2` *AIC for an xreg2 Model*

Description

Computes the Akaike Information Criterion for a fitted `xreg2` model. Scale parameters (SIGMA, LN_SIGMA, OMEGA, LN_OMEGA) are excluded from the parameter count.

Usage

```
## S3 method for class 'xreg2'
AIC(object, ..., k = 2, just_total = TRUE)
```

Arguments

`object`	A fitted `xreg2` object.
`...`	Currently unused.
`k`	Numeric; penalty per parameter. Default `2`.
`just_total`	Logical; return only the total AIC across all components. Default `TRUE`.

Details

AIC method for xreg2 objects

Value

A numeric scalar (AIC value).

`ARE` *Average Relative Error*

Description

Average Relative Error

Usage

```
ARE(est, obs, absolute = FALSE, ...)

## S3 method for class 'data.frame'
ARE(est, obs, absolute = FALSE, ...)

## S3 method for class 'matrix'
ARE(est, obs, absolute = FALSE, ...)

## S3 method for class 'numeric'
ARE(est, obs, absolute = FALSE, ...)

ARE_conform(dfys, dfobs, absolute = FALSE)
```

Arguments

est	Numeric; estimates.
obs	Numeric; observations.
absolute	Logical; return absolute version. Default FALSE.
...	Currently unused.
dfys	Numeric matrix or data frame of estimates.
dfobs	Numeric matrix or data frame of observations.

Value

Numeric ARE value.

BIC.xreg2 *BIC for an xreg2 Model*

Description

Computes the Bayesian Information Criterion for a fitted xreg2 model.

Usage

```
## S3 method for class 'xreg2'
BIC(object, ..., just_total = TRUE)
```

Arguments

object	A fitted xreg2 object.
...	Currently unused.
just_total	Logical; return only the total BIC. Default TRUE.

Details

BIC method for xreg2 objects

Value

A numeric scalar (BIC value).

---

c.xreg2Control *Combine xreg2 Control Objects*

Description

Combines one or more `xreg2Control` objects into an `xreg2ControlList` suitable for passing to `xreg2_fit`.

Usage

```
## S3 method for class 'xreg2Control'
c(...)

## S3 method for class 'xreg2ControlList'
c(...)
```

Arguments

`...`	One or more `xreg2Control` objects. Named arguments are used as component names; unnamed arguments use the object's `name` field or a sequential integer.

Details

Combine xreg2Control objects into a list

Value

An object of class `c("xreg2ControlList", "list")`.

See Also

`xreg2_control`, `xreg2_fit`

Examples

```
ctrl1 <- xreg2_control(formulas = list(y ~ INTERCEPT + x * SLOPE),
                       name = "component1")
cl    <- c(ctrl1)
```

combmean *Combined mean of two groups*

Description

Combined mean of two groups

Usage

```
combmean(m1, m2, n1 = NULL, n2 = NULL, na.rm = FALSE)
```

Arguments

m1	Mean of group 1 (scalar or vector).
m2	Mean of group 2 (scalar or vector).
n1	Size of group 1.
n2	Size of group 2.
na.rm	Logical.

Value

Combined mean.

combmeans *Combined mean of multiple groups*

Description

Combined mean of multiple groups

Usage

```
combmeans(ms, ns, na.rm = FALSE)
```

Arguments

ms	Numeric vector of group means.
ns	Numeric vector of group sizes.
na.rm	Logical.

Value

Combined mean.

combmssds *Combined mean, SD and n for multiple groups*

Description

Combined mean, SD and n for multiple groups

Usage

```
combmssds(ms, ss, ns, na.rm = FALSE)
```

Arguments

ms	Numeric vector of group means.
ss	Numeric vector of group SDs.
ns	Numeric vector of group sizes.
na.rm	Logical.

Value

Matrix with columns n, mean, sd.

combsd *Combined standard deviation of two groups*

Description

Combined standard deviation of two groups

Usage

```
combsd(m1, m2, s1, s2, n1, n2, na.rm = FALSE)
```

Arguments

m1	Mean of group 1.
m2	Mean of group 2.
s1	SD of group 1.
s2	SD of group 2.
n1	Size of group 1.
n2	Size of group 2.
na.rm	Logical.

Value

Combined SD.

combsds *Combined SD of multiple groups*

Description

Combined SD of multiple groups

Usage

```
combsds(ms, ss, ns, na.rm = FALSE)
```

Arguments

ms	Numeric vector of group means.
ss	Numeric vector of group SDs.
ns	Numeric vector of group sizes.
na.rm	Logical.

Value

Combined SD.

comp_table *Comparison Table*

Description

Computes multiple error metrics and returns them as a matrix.

Usage

```
comp_table(est, obs, ...)
```

Arguments

est	Numeric vector or matrix of estimates.
obs	Numeric vector of observations.
...	Passed to metric functions.

Details

Comparison table of error metrics

Value

A numeric matrix with one row per metric.

erMAX *Maximum Absolute Relative Error*

Description

Maximum Absolute Relative Error

Usage

```
erMAX(est, obs, ...)

## S3 method for class 'data.frame'
erMAX(est, obs, ...)

## S3 method for class 'matrix'
erMAX(est, obs, ...)

## S3 method for class 'numeric'
erMAX(est, obs, ...)
```

Arguments

est	Numeric; estimates.
obs	Numeric; observations.
...	Currently unused.

Value

Numeric erMAX value.

fisher_icc *Fisher's ICC*

Description

Computes Fisher's intraclass correlation coefficient between two numeric vectors or matrices.

Usage

```
fisher_icc(y, x, ...)

## S3 method for class 'data.frame'
fisher_icc(y, x, ...)

## S3 method for class 'matrix'
```

```
fisher_icc(y, x, ...)

## S3 method for class 'numeric'
fisher_icc(y, x, ...)
```

Arguments

y	Numeric vector, matrix, or data frame.
x	Numeric vector, matrix, or data frame.
...	Currently unused.

Details

Fisher's intraclass correlation coefficient

Value

Numeric ICC value.

---

fixdist *Standardise Distribution Name*

Description

Maps common distribution name variants to their canonical R short form (e.g., `"normal"` -> `"norm"`).

Usage

```
fixdist(dist)
```

Arguments

dist	Character distribution name.

Details

Standardise distribution name

Value

Canonical short-form character name.

getf *Get Distribution Function*

Description

Returns the named R distribution function (density, CDF, quantile, or random) for a given distribution, with an extra ... argument appended to its formals.

Usage

```
getf(dist, type)

getp(dist)

getq(dist)

getd(dist)

getr(dist)
```

Arguments

dist	Character distribution name (passed to fixdist).
type	Character; one of "p", "d", "q", "r".

Details

Get a distribution function by type

Value

A function.

getMean *Mean of a distribution*

Description

Mean of a distribution

Usage

```
getMean(qfun, pars)
```

Arguments

qfun	Quantile function.
pars	Named list of distribution parameters.

Value

Numeric mean.

---

getSD *Standard deviation of a distribution*

Description

Standard deviation of a distribution

Usage

```
getSD(qfun, pars)
```

Arguments

qfun	Quantile function or character distribution name.
pars	Named list of distribution parameters.

Value

Numeric standard deviation.

---

getVar *Variance of a distribution*

Description

Variance of a distribution

Usage

```
getVar(qfun, pars)
```

Arguments

qfun	Quantile function or character distribution name.
pars	Named list of distribution parameters.

Value

Numeric variance.

is.wholenumber *Test for Whole Number*

Description

Tests whether numeric values are whole numbers within a numeric tolerance.

Usage

```
is.wholenumber(x, tol = .Machine$double.eps^0.5)
```

Arguments

x	Numeric vector.
tol	Numeric tolerance. Default .Machine$double.eps^0.5.

Details

Test for whole number

Value

Logical vector.

just_CCC *Lin's CCC (simplified)*

Description

Lin's concordance correlation coefficient (simplified)

Usage

```
just_CCC(y, x, ...)

## S3 method for class 'data.frame'
just_CCC(y, x, ...)

## S3 method for class 'matrix'
just_CCC(y, x, ...)

## S3 method for class 'numeric'
just_CCC(y, x, ...)
```

Arguments

y	Numeric vector, matrix, or data frame.
x	Numeric vector, matrix, or data frame.
...	Passed to LCCC.

Value

Numeric CCC value.

LCCC *Lin's Concordance Correlation Coefficient*

Description

Calculates Lin's concordance correlation coefficient (CCC) for a pair of variables, measuring agreement along the 45-degree line.

Usage

```
LCCC(
  x,
  y,
  ci = "z-transform",
  conf.level = 0.95,
  na.rm = FALSE,
  justp = FALSE,
  ...
)
```

Arguments

x	Numeric vector; first variable.
y	Numeric vector; second variable.
ci	Character; method for confidence interval. Default "z-transform".
conf.level	Numeric; confidence level. Default 0.95.
na.rm	Logical; remove NA values. Default FALSE.
justp	Logical; if TRUE return only the point estimate. Default FALSE.
...	Currently unused.

Details

Lin's concordance correlation coefficient

Value

A list with components rho.c, s.shift, l.shift, C.b, and blalt.

MAE *Mean Absolute Error*

Description

Mean Absolute Error

Usage

```
MAE(est, obs, ...)

## S3 method for class 'data.frame'
MAE(est, obs, ...)

## S3 method for class 'matrix'
MAE(est, obs, ...)

## S3 method for class 'numeric'
MAE(est, obs, ...)
```

Arguments

est	Numeric; estimates.
obs	Numeric; observations.
...	Passed to mean.

Value

Numeric MAE value.

MAPE *Mean Absolute Relative Error (MARE / MAPE)*

Description

Mean Absolute Relative Error / Mean Absolute Percentage Error

Usage

```
MAPE(est, obs, ...)

MARE(est, obs, ...)

## S3 method for class 'data.frame'
MARE(est, obs, ...)
```

```
## S3 method for class 'matrix'
MARE(est, obs, ...)

## S3 method for class 'numeric'
MARE(est, obs, ...)
```

Arguments

est	Numeric; estimates.
obs	Numeric; observations.
...	Currently unused.

Value

Numeric MARE value.

MBE *Mean Bias Error*

Description

Mean Bias Error

Usage

```
MBE(est, obs, ...)

## S3 method for class 'data.frame'
MBE(est, obs, ...)

## S3 method for class 'matrix'
MBE(est, obs, ...)

## S3 method for class 'numeric'
MBE(est, obs, ...)
```

Arguments

est	Numeric; estimates.
obs	Numeric; observations.
...	Passed to mean.

Value

Numeric MBE value.

ordD *Order Statistic Density*

Description

Computes the density of the r-th order statistic in a random sample of n from a given distribution.

Usage

```
ordD(x, r, n, par = c(mean = 0, sd = 1), dist = "norm", log = FALSE)
```

Arguments

x	Numeric vector of values.
r	Integer rank.
n	Sample size.
par	Named vector of distribution parameters.
dist	Character distribution name.
log	Logical; return log-density. Default FALSE.

Details

Density for an order statistic

Value

Numeric density (or log-density).

ordE *Order Statistic Expected Value*

Description

Computes the expected value of the r-th order statistic in a random sample of n from a given distribution.

Usage

```
ordE(r, n, par = c(mean = 0, sd = 1), dist = "norm", log = FALSE)
```

Arguments

r	Integer rank (or vector of ranks).
n	Sample size.
par	Named vector of distribution parameters.
dist	Character distribution name.
log	Currently unused.

Details

Expected value of an order statistic

Value

Numeric expected value.

---

ordEp *Order Statistic Expected Probabilities*

Description

Expected probability of order statistics

Usage

```
ordEp(probs, n, par = c(mean = 0, sd = 1), dist = "normal")
```

Arguments

probs	Numeric vector of rank proportions.
n	Sample size.
par	Named vector of distribution parameters.
dist	Character distribution name.

Value

Numeric vector of probabilities.

predict.xreg2 *Predict from an xreg2 Model*

Description

Generates predictions (linear predictors and intermediate computed variables) from a fitted xreg2 model applied to new data.

Usage

```
## S3 method for class 'xreg2'
predict(object, newdata = NULL, return_vector = FALSE, ...)
```

Arguments

object	A fitted xreg2 object.
newdata	A named list of data frames (one per model component), or a single data frame (automatically wrapped and named after the first component).
return_vector	Logical; if TRUE and there is a single component, return a numeric vector of Xb values instead of a list. Default FALSE.
...	Currently unused.

Details

Predict method for xreg2 objects

Value

A named list of data frames (one per component), each augmented with computed variables including Xb.

Examples

```
set.seed(1)
df   <- data.frame(y = rnorm(200, 2, 0.5), x = rnorm(200))
ctrl <- xreg2_control(
  formulas     = list(y ~ INTERCEPT + x * SLOPE),
  start_values = c(INTERCEPT = 0, SLOPE = 0),
  name         = "main"
)
fit  <- xreg2_fit(controlList = c(ctrl), dataList = list(main = df))
newdf <- data.frame(x = 1:5)
preds <- predict(fit, newdata = list(main = newdf))
```

print.xreg2 *Print an xreg2 Model*

Description

Prints a concise summary of a fitted xreg2 model, including coefficients, standard errors, and minimised negative log-likelihoods.

Usage

```
## S3 method for class 'xreg2'
print(x, ...)
```

Arguments

x	A fitted xreg2 object.
...	Currently unused.

Details

Print method for xreg2 objects

Value

x invisibly.

quant_fun *Quantile Summary*

Description

Computes summary statistics and quantiles for a numeric vector.

Usage

```
quant_fun(
  x,
  statfuns = function(y) c(MEAN = mean(y, na.rm = TRUE), SE = sd(y, na.rm = TRUE)),
  probs = c(min = 0, `2.5%` = 0.025, `25%` = 0.25, median = 0.5, `75%` = 0.75,
    `97.5%` = 0.975, max = 1),
  names = TRUE
)
```

Arguments

x	Numeric vector.
statfuns	Function returning named summary statistics.
probs	Named numeric vector of quantile probabilities.
names	Logical; include names in quantile output.

Details

Quantile summary statistics

Value

Named numeric vector of statistics and quantiles.

---

quant_fun_groups *Quantile Summaries by Group*

Description

Computes quantile summaries within each level of a categorical variable.

Usage

```
quant_fun_groups(x, catvar, pooled = TRUE, n = TRUE)

quant_funs_groups(x, catvar, pooled = TRUE, n = TRUE)
```

Arguments

x	A list of numeric vectors.
catvar	Factor or character vector of group labels.
pooled	Logical; include a pooled row. Default TRUE.
n	Logical; include count and missing count columns. Default TRUE.

Details

Quantile summaries by group

Value

Data frame with one row per group.

quant_funs *Quantile Summaries (Matrix)*

Description

Applies quant_fun over rows or columns of a matrix or data frame.

Usage

```
quant_funs(
  x,
  MARGIN = 1,
  statfuns = function(y) c(MEAN = mean(y, na.rm = TRUE), SE = sd(y, na.rm = TRUE)),
  probs = c(min = 0, `2.5%` = 0.025, `25%` = 0.25, median = 0.5, `75%` = 0.75,
    `97.5%` = 0.975, max = 1)
)
```

Arguments

x	Matrix or data frame.
MARGIN	Integer; 1 for rows, 2 for columns.
statfuns	Function returning named summary statistics.
probs	Named numeric vector of quantile probabilities.

Details

Quantile summaries over matrix margins

Value

Data frame of statistics.

quant_funst *Transposed Quantile Summaries (Matrix)*

Description

Like quant_funs but returns transposed result.

Usage

```
quant_funst(
  x,
  MARGIN = 1,
  statfuns = function(y) c(MEAN = mean(y, na.rm = TRUE), SE = sd(y, na.rm = TRUE)),
  probs = c(min = 0, `2.5%` = 0.025, `25%` = 0.25, median = 0.5, `75%` = 0.75,
    `97.5%` = 0.975, max = 1)
)
```

Arguments

x	Matrix or data frame.
MARGIN	Integer; 1 for rows, 2 for columns.
statfuns	Function returning named summary statistics.
probs	Named numeric vector of quantile probabilities.

Details

Transposed quantile summaries over matrix margins

Value

Transposed data frame of statistics.

ReMSE *Relative Mean Squared Error*

Description

Relative Mean Squared Error

Usage

```
ReMSE(est, obs, trueobs, ...)

## S3 method for class 'data.frame'
ReMSE(est, obs, trueobs, ...)

## S3 method for class 'matrix'
ReMSE(est, obs, trueobs, ...)

## S3 method for class 'numeric'
ReMSE(est, obs, trueobs, ...)

ReMSE_conform(dfys, dfobs, trueobs)
```

Arguments

`est`	Numeric; estimates.
`obs`	Numeric; observations.
`trueobs`	Numeric; true values.
`...`	Currently unused.
`dfys`	Numeric matrix or data frame of estimates.
`dfobs`	Numeric matrix or data frame of observations.

Value

Numeric ReMSE value.

`RMSE` *Root Mean Square Error*

Description

Root Mean Square Error

Usage

```
RMSE(est, obs, percentage = FALSE, ...)

## S3 method for class 'data.frame'
RMSE(est, obs, percentage = FALSE, ...)

## S3 method for class 'matrix'
RMSE(est, obs, percentage = FALSE, ...)

## S3 method for class 'numeric'
RMSE(est, obs, percentage = FALSE, ...)

pRMSE(est, obs, ...)

rRMSE(est, obs, ...)
```

Arguments

`est`	Numeric; estimates.
`obs`	Numeric; observations.
`percentage`	Logical; return as percentage of mean. Default `FALSE`.
`...`	Passed to `RMSE`.

Value

Numeric RMSE value.

RMSRE *Root Mean Square Relative Error*

Description

Root Mean Square Relative Error

Usage

```
RMSRE(est, obs, ...)

## S3 method for class 'data.frame'
RMSRE(est, obs, ...)

## S3 method for class 'matrix'
RMSRE(est, obs, ...)

## S3 method for class 'numeric'
RMSRE(est, obs, ...)
```

Arguments

est	Numeric; estimates.
obs	Numeric; observations.
...	Currently unused.

Value

Numeric RMSRE value.

summary.xreg2 *Summarise an xreg2 Model*

Description

Prints a detailed summary of a fitted xreg2 model. Currently equivalent to print.xreg2.

Usage

```
## S3 method for class 'xreg2'
summary(object, ...)
```

Arguments

object	A fitted xreg2 object.
...	Currently unused.

Details

Summary method for xreg2 objects

Value

`object` invisibly.

`t_test2` *Two-Sample t-Test from Summary Statistics*

Description

Two-sample t-test from summary statistics

Usage

```
t_test2(m1, m2, s1, s2, n1, n2, m0 = 0, equal.variance = FALSE)
```

Arguments

`m1`	Numeric; mean of sample 1.
`m2`	Numeric; mean of sample 2.
`s1`	Numeric; SD of sample 1.
`s2`	Numeric; SD of sample 2.
`n1`	Numeric; size of sample 1.
`n2`	Numeric; size of sample 2.
`m0`	Numeric; null difference. Default `0`.
`equal.variance`	Logical; assume equal variances. Default `FALSE`.

Value

Named numeric vector with test statistics and confidence interval.

tstat *t-Statistic*

Description

t-statistic for model comparison

Usage

```
tstat(est, obs, ...)

## S3 method for class 'data.frame'
tstat(est, obs, ...)

## S3 method for class 'matrix'
tstat(est, obs, ...)

## S3 method for class 'numeric'
tstat(est, obs, ...)
```

Arguments

est	Numeric; estimates.
obs	Numeric; observations.
...	Currently unused.

Value

Numeric t-statistic.

U95 *Uncertainty at 95 Percent*

Description

Uncertainty at 95 Percent

Usage

```
U95(est, obs, ...)

## S3 method for class 'data.frame'
U95(est, obs, ...)

## S3 method for class 'matrix'
```

```
U95(est, obs, ...)

## S3 method for class 'numeric'
U95(est, obs, ...)
```

Arguments

est	Numeric; estimates.
obs	Numeric; observations.
...	Passed to var and sqrt.

Value

Numeric U95 value.

---

xreg2_cont *General Continuous Negative Log-Likelihood*

Description

Computes the negative log-likelihood for a censored/interval regression model supporting multiple distributions (normal, lognormal, exponential, beta, gamma, weibull). Designed as a p_fun argument for xreg2_control.

Usage

```
xreg2_cont(
  par,
  dtalist,
  log.p = TRUE,
  aggregate.p = TRUE,
  print_pars = FALSE,
  dist = "normal"
)
```

Arguments

par	Named numeric vector of current parameter values.
dtalist	Internal data list produced by xreg2_fit.
log.p	Logical; use log-scale probabilities. Default TRUE.
aggregate.p	Logical; return scalar negloglik. Default TRUE.
print_pars	Logical; print parameters and value. Default FALSE.
dist	Character; distribution family. One of "normal", "lognormal", "exp"/"exponential", "beta", "gamma", "weibull". Default "normal".

Details

General continuous log-likelihood for xreg2

Value

Scalar negative log-likelihood (when `aggregate.p = TRUE`).

Examples

```
# ctrl <- xreg2_control(formulas = list(y ~ INTERCEPT + x * SLOPE),
#                       p_fun = xreg2_cont,
#                       start_values = c(INTERCEPT = 0, SLOPE = 0, LN_SIGMA = 0))
```

---

xreg2_cont_normal *Continuous Normal Negative Log-Likelihood*

Description

Computes the negative log-likelihood for a censored/interval normal regression model. Designed as a `p_fun` argument for `xreg2_control`.

Usage

```
xreg2_cont_normal(
  par,
  dtalist,
  log.p = TRUE,
  aggregate.p = TRUE,
  print_pars = FALSE
)
```

Arguments

`par`	Named numeric vector of current parameter values.
`dtalist`	Internal data list produced by `xreg2_fit`.
`log.p`	Logical; use log-scale probabilities. Default `TRUE`.
`aggregate.p`	Logical; return scalar negloglik. Default `TRUE`.
`print_pars`	Logical; print parameters and value. Default `FALSE`.

Details

Continuous normal log-likelihood for xreg2

Value

Scalar negative log-likelihood (when `aggregate.p = TRUE`).

Examples

```
# Typically used as p_fun in xreg2_control():
# ctrl <- xreg2_control(formulas = list(y ~ INTERCEPT + x * SLOPE),
#                       p_fun = xreg2_cont_normal)
```

`xreg2_cont_r_lnorm` *Continuous Log-Normal Negative Log-Likelihood with Random Intercept*

Description

Computes the negative log-likelihood for a mixed-effects log-normal regression model with a random intercept, integrated via Gauss-Hermite quadrature. Designed as a `p_fun` argument for `xreg2_control`.

Usage

```
xreg2_cont_r_lnorm(par, dtalist)
```

Arguments

`par`	Named numeric vector of current parameter values. Must include either `LN_OMEGA` or `OMEGA`.
`dtalist`	Internal data list produced by `xreg2_fit`. Must include an `internal_id` column.

Details

Continuous log-normal log-likelihood with random intercept

Value

Scalar negative log-likelihood.

Examples

```
# ctrl <- xreg2_control(formulas = list(y ~ INTERCEPT + x * SLOPE),
#                       p_fun = xreg2_cont_r_lnorm)
```

xreg2_cont_r_normal *Continuous Normal Negative Log-Likelihood with Random Intercept*

Description

Computes the negative log-likelihood for a mixed-effects normal regression model with a random intercept, integrated via Gauss-Hermite quadrature. Designed as a `p_fun` argument for `xreg2_control`.

Usage

```
xreg2_cont_r_normal(par, dtalist, print_pars = FALSE)
```

Arguments

`par`	Named numeric vector of current parameter values. Must include either `LN_OMEGA` or `OMEGA` (between-subject SD).
`dtalist`	Internal data list produced by `xreg2_fit`. Must include an `internal_id` column.
`print_pars`	Logical; print parameters and value. Default `FALSE`.

Details

Continuous normal log-likelihood with random intercept

Value

Scalar negative log-likelihood.

Examples

```
# ctrl <- xreg2_control(formulas = list(y ~ INTERCEPT + x * SLOPE),
#                       p_fun = xreg2_cont_r_normal)
```

xreg2_control *Create xreg2 Control Object*

Description

Creates a control object specifying the model formula(s), likelihood function, gradient function, and starting values for one component of a joint `xreg2` model.

Usage

```
xreg2_control(
  formulas,
  start_values = numeric(),
  fixed_values = numeric(),
  p_fun = xreg2_cont_normal,
 grad_fun = function(par, ...) numDeriv::grad(func = p_fun, x = par, method = "simple",
    ...),
  p_aggregation_fun = function(d_df) return(-d_df$p * d_df$internal_count),
  weights_var = NA,
  name = NA,
  censor_bounds = c(-Inf, Inf),
  lower = NA,
  upper = NA
)
```

Arguments

formulas	A formula or list of formulas. The left-hand side names the dependent (or intermediate) variable; the right-hand side is evaluated in the context of parameter values and data columns.
start_values	Named numeric vector of starting values. An existing xreg2 fitted object may be supplied; coefficients are extracted.
fixed_values	Named numeric vector of parameter values held fixed during optimisation.
p_fun	Log-likelihood function. Default xreg2_cont_normal. Must accept (par, dtalist, ...) and return a scalar.
grad_fun	Gradient function. Default uses numDeriv::grad with method "simple".
p_aggregation_fun	Aggregation function applied to the likelihood data-frame. Rarely changed by users.
weights_var	Character name of a column in the data used as observation weights. NA for unweighted.
name	Optional character label for this component.
censor_bounds	Numeric vector of length 2 giving lower and upper censoring bounds. Default c(-Inf, Inf) (no censoring).
lower	Named numeric vector of lower parameter bounds (for bounded optimisers).
upper	Named numeric vector of upper parameter bounds.

Details

Create an xreg2 control object

Value

An object of class c("xreg2Control", "list").

See Also

xreg2_fit, c.xreg2Control

Examples

```
ctrl <- xreg2_control(
  formulas     = list(y ~ INTERCEPT + x * SLOPE),
  start_values = c(INTERCEPT = 0, SLOPE = 0),
  p_fun        = xreg2_cont_normal,
  name         = "continuous"
)
```

---

xreg2_dich_logistic *Dichotomous Logistic Negative Log-Likelihood*

Description

Computes the negative log-likelihood for a binary logistic regression model. Designed as a p_fun argument for xreg2_control.

Usage

```
xreg2_dich_logistic(par, dtalist)
```

Arguments

par	Named numeric vector of current parameter values.
dtalist	Internal data list produced by xreg2_fit.

Details

Dichotomous logistic log-likelihood for xreg2

Value

Scalar negative log-likelihood.

Examples

```
# ctrl <- xreg2_control(formulas = list(y ~ INTERCEPT + x * SLOPE),
#                       p_fun = xreg2_dich_logistic,
#                       start_values = c(INTERCEPT = 0, SLOPE = 0))
```

xreg2_fit *Fit xreg2 Model*

Description

Fits a flexible maximum-likelihood regression model supporting censored, interval, and hybrid data. Accepts a list of xreg2_control objects and a matching named list of data frames.

Usage

```
xreg2_fit(
  controlList,
  dataList = NULL,
  start_values = numeric(),
  fixed_values = numeric(),
  latent_classes = 0,
  latent_class_parameters = character(),
  latent_id_colname = character(),
  return_type = "fit",
  print_sum = FALSE,
  method = "ucminf",
  hessian = TRUE,
  run_from = NULL,
  ...
)
```

Arguments

controlList	An xreg2ControlList (produced by combining xreg2_control objects with c()), an xreg2_obj (for re-optimisation), or an xreg2/xreg fitted object (for re-fitting).
dataList	Named list of data frames, one per component. Names must match those of controlList.
start_values	Optional named numeric vector of starting values, overriding those in controlList.
fixed_values	Named numeric vector of parameter values held fixed.
latent_classes	Integer; number of latent classes. 0 disables.
latent_class_parameters	Character vector of parameter names that vary by latent class.
latent_id_colname	Character name of the grouping column for latent classes.
return_type	Character; "fit" (default), "control", "df", or "predict".
print_sum	Logical; print summary during optimisation. Default FALSE.
method	Character; optimisation method. Default "ucminf".
hessian	Logical; compute Hessian for standard errors. Default TRUE.

run_from	Internal; character string identifying the calling wrapper (e.g. "hyreg"). Used to suppress certain warnings when called from higher-level functions.
...	Additional arguments passed to the optimiser.

Details

Fit a flexible maximum-likelihood regression model

Value

An object of class "xreg2" with fitted coefficients, standard errors, log-likelihoods, and counts.

See Also

xreg2_control, xreg2_optim, predict.xreg2

Examples

```
set.seed(1)
df   <- data.frame(y = rnorm(200, 2, 0.5), x = rnorm(200))
ctrl <- xreg2_control(
  formulas     = list(y ~ INTERCEPT + x * SLOPE),
  start_values = c(INTERCEPT = 0, SLOPE = 0),
  name         = "main"
)
fit  <- xreg2_fit(controlList = c(ctrl), dataList = list(main = df))
coef(fit)
```

---

xreg2_fit_s *Fit xreg2 Model from Pre-Structured Data*

Description

Simplified version of xreg2_fit that accepts data already containing ub (upper bound) and lb (lower bound) columns instead of requiring data pre-processing.

Usage

```
xreg2_fit_s(
  controlList,
  dataList = NULL,
  start_values = numeric(),
  fixed_values = numeric(),
  latent_classes = 0,
  latent_class_parameters = character(),
  latent_id_colname = character(),
```

```
  return_type = "fit",
  print_sum = FALSE,
  method = "ucminf",
  hessian = TRUE,
  ...
)
```

Arguments

controlList	An xreg2ControlList (produced by combining xreg2_control objects with c()), an xreg2_obj (for re-optimisation), or an xreg2/xreg fitted object (for re-fitting).
dataList	Named list of data frames, one per component. Names must match those of controlList.
start_values	Optional named numeric vector of starting values, overriding those in controlList.
fixed_values	Named numeric vector of parameter values held fixed.
latent_classes	Integer; number of latent classes. 0 disables.
latent_class_parameters	Character vector of parameter names that vary by latent class.
latent_id_colname	Character name of the grouping column for latent classes.
return_type	Character; "fit" (default), "control", "df", or "predict".
print_sum	Logical; print summary during optimisation. Default FALSE.
method	Character; optimisation method. Default "ucminf".
hessian	Logical; compute Hessian for standard errors. Default TRUE.
...	Additional arguments passed to the optimiser.

Details

Fit an xreg2 model from pre-structured data

Value

An object of class "xreg2".

See Also

xreg2_fit, xreg2_control

Examples

```
set.seed(1)
# xreg2_fit_s expects pre-structured data with lb and ub columns.
# Use lb == ub for exact (uncensored) observations.
y    <- rnorm(100, 2, 0.5)
df   <- data.frame(y = y, x = rnorm(100), lb = y, ub = y)
ctrl <- xreg2_control(
```

```
  formulas     = list(y ~ INTERCEPT + x * SLOPE),
  start_values = c(INTERCEPT = 0, SLOPE = 0, LN_SIGMA = 0),
  name         = "main"
)
fit  <- xreg2_fit_s(controlList = c(ctrl), dataList = list(main = df))
```

`xreg2_hyreg` *Hybrid Regression (xreg2 version)*

Description

High-level wrapper around `xreg2_fit` replicating the interface of the STATA `hyreg` command and `xreg::hyreg()`. Simultaneously fits a censored continuous model (normal or logistic) and a dichotomous model (logistic or normal) to a single data frame that contains both observation types.

Usage

```
xreg2_hyreg(
  formula,
  df,
  datatype = "method",
  init = numeric(),
  contdist = "normal",
  dichdist = "logistic",
  hetcont = NULL,
  hetdich = NULL,
  ul = Inf,
  ll = -Inf,
  lntheta = TRUE,
  lnsigma = TRUE,
  dichformula = NULL,
  ...
)
```

Arguments

`formula`	A formula or list of formulas for the continuous component. The dependent variable (left-hand side) must match a column in `df`, or use interval columns named `varname.lb` and `varname.ub`.
`df`	A `data.frame` containing both continuous and dichotomous observations. Must include the column specified by `datatype`.
`datatype`	Character; name of the column that distinguishes continuous (`TRUE`) from dichotomous (`FALSE`) observations. Default `"method"`.
`init`	Optional named numeric vector (or `xreg2` object) of starting values.

contdist	Character; distribution for the continuous component. "normal" (default) or "logistic".
dichdist	Character; distribution for the dichotomous component. "logistic" (default) or "normal".
hetcont	Optional formula for heteroscedastic standard deviations in the continuous component. If NULL (default), uses sigma_est ~ SIGMA (or sigma_est ~ exp(SIGMA) when lnsigma = TRUE).
hetdich	Optional formula for heteroscedastic standard deviations in the dichotomous component. If NULL (default), uses theta_est ~ THETA (or theta_est ~ exp(THETA) when lntheta = TRUE).
ul	Numeric; upper censoring bound. Default Inf.
ll	Numeric; lower censoring bound. Default -Inf.
lntheta	Logical; exponentiate theta prior to fitting (matches STATA default). Default TRUE.
lnsigma	Logical; exponentiate sigma prior to fitting (matches STATA default). Default TRUE.
dichformula	Optional formula for the dichotomous component. Defaults to formula.
...	Additional arguments forwarded to xreg2_fit.

Details

Hybrid regression wrapper for xreg2

Value

An object of class "xreg2".

See Also

xreg2_fit, xreg2_control

Examples

```
set.seed(1)
n   <- 300
df  <- data.frame(
  value  = c(rnorm(200, 0.6, 0.3), rep(NA, 100)),
  age    = rnorm(n),
  method = c(rep(TRUE, 200), rep(FALSE, 100))
)
df$value[df$method == FALSE] <- as.integer(df$age[df$method == FALSE] > 0)

fit <- xreg2_hyreg(
  formula  = value ~ INTERCEPT + age * AGE,
  df       = df,
  datatype = "method"
)
print(fit)
```

xreg2_optim *Optimise an xreg2 Model*

Description

Low-level optimisation routine called by xreg2_fit and xreg2_fit_s. Can also be called directly on an xreg2_obj to re-optimise with different settings.

Optimizer routing:

- "ucminf" (default) — uses **ucminf** with central-difference gradients and high-precision tolerances.
- "L-BFGS-B", "BFGS", "CG", "Nelder-Mead", "SANN", "Brent" — uses stats::optim(). Box constraints (lower/upper set in xreg2_control) are enforced for "L-BFGS-B".
- Any other method string — requires the **optimx** package (listed in Suggests) and dispatches via optimx::optimr().

Method-specific control defaults (high-precision) are set automatically and can be overridden by passing optim_control = list(...) to xreg2_fit.

Usage

```
xreg2_optim(
  xreg2obj,
  method = NULL,
  hessian = TRUE,
  fixed_values = NULL,
  optim_control = NULL,
  ...
)
```

Arguments

xreg2obj	An xreg2_obj list produced by xreg2_fit with return_type = "control".
method	Character; optimisation method. NULL defaults to "ucminf".
hessian	Logical; compute Hessian. Default TRUE.
fixed_values	Optional named numeric vector of additional fixed values.
optim_control	Optional named list of control parameters that override the method-specific defaults set by xreg2_fit. For example, optim_control = list(maxit = 500) tightens the iteration budget. Passed directly to ucminf, optim, or optimx::optimr depending on method.
...	Additional arguments passed to the optimiser objective function.

Details

Optimise an xreg2 model object

Value

An object of class "xreg2".

See Also

xreg2_fit

Examples

```
set.seed(1)
df   <- data.frame(y = rnorm(200, 2, 0.5), x = rnorm(200))
ctrl <- xreg2_control(
  formulas     = list(y ~ INTERCEPT + x * SLOPE),
  start_values = c(INTERCEPT = 0, SLOPE = 0),
  name         = "main"
)
obj          <- xreg2_fit(controlList = c(ctrl), dataList = list(main = df),
                          return_type = "control")
fit          <- xreg2_optim(obj, method = "ucminf")
fit_lbfgsb   <- xreg2_optim(obj, method = "L-BFGS-B")
```

Index