Package 'glmx'

September 4, 2024

Version 0.2-1

Date 2024-09-03

Title Generalized Linear Models Extended

Description
Extended techniques for generalized linear models (GLMs), especially for binary responses, including parametric links and heteroscedastic latent variables.

Depends R (>= 3.0.0)

Imports stats, MASS, Formula, lmtest, sandwich

Suggests AER, gld, numDeriv, pscl

License GPL-2 | GPL-3

NeedsCompilation no

Author Achim Zeileis [aut, cre] (<https://orcid.org/0000-0003-0918-3766>),
Roger Koenker [aut] (<https://orcid.org/0000-0001-8436-6306>),
Philipp Doebler [aut] (<https://orcid.org/0000-0002-2946-8526>)

Maintainer Achim Zeileis <Achim.Zeileis@R-project.org>

Repository CRAN

Date/Publication 2024-09-04 00:10:03 UTC

Contents

`AbortionAmbivalence` *American Ambivalence towards Abortion Policy*

Description

Data about attitudes towards abortion policy in the US. Cross-section data from the US General Social Survey 1982 with oversample of African American respondents.

Usage

```
data("AbortionAmbivalence")
```

Format

A data frame containing 1860 observations on 20 variables.

health factor. Answer to the question: Please tell me whether or not you think it should be possible for a pregnant woman to obtain a legal abortion if the woman's own health is seriously endangered by the pregnancy?

rape factor. Answer to the question: Please tell me whether or not you think it should be possible for a pregnant woman to obtain a legal abortion if she became pregnant as a result of rape?

defect factor. Answer to the question: Please tell me whether or not you think it should be possible for a pregnant woman to obtain a legal abortion if there is a strong chance of serious defect in the baby?

poor factor. Answer to the question: Please tell me whether or not you think it should be possible for a pregnant woman to obtain a legal abortion if the family has a very low income and cannot afford any more children?

nomore factor. Answer to the question: Please tell me whether or not you think it should be possible for a pregnant woman to obtain a legal abortion if she is married and does not want any more children?

single factor. Answer to the question: Please tell me whether or not you think it should be possible for a pregnant woman to obtain a legal abortion if she is not married and does not want to marry the man?

any factor. Answer to the question: Please tell me whether or not you think it should be possible for a pregnant woman to obtain a legal abortion if the woman wants it for reason?

ethnicity factor indicating ethnicity. Is the individual African-American (`"afam"`) or not (`"other"`)?

gender factor indicating gender.

religion factor indicating religious preference (`"catholic"` or `"other"`).

religiousness Religious intensity as coded by Alvarez and Brehm (1995).

religiousness2 Religious intensity in an alternative coding suggested by Altman and McDonald (1995).

church Numeric coding of frequency of attending church.

erameans factor. Answer to the question: Do you understand what the Equal Rights Amendment (ERA) means?

erasupport Intensity of support for ERA.

pros Number of arguments in favor of abortion named by the subject.

cons Number of arguments against abortion named by the subject.

importance Numeric coding of subjective importance of abortion issue.

information Numeric coding of self-assessment of information on abortion issue available to the subject.

firmness Numeric coding of subjective firmness of opinion on abortion.

Details

The data were prepared and analyzed by Alvarez and Brehm (1995). A detailed discussion of the variables is provided in their Appendix A and the model is developed in their Section 3.

The data were reanalyzed by Altman and McDonald (2003) with focus on numerical accuracy and by Keele and Park (2006) with focus on interpretability.

Source

Online supplements to Altman and McDonald (2003).

doi:10.1093/pan/mpg016

References

Altman M, McDonald MP (2003). "Replication with Attention to Numerical Accuracy." *Political Analysis*, **11**, 302–307.

Alvarez RM, Brehm J (1995). "American Ambivalence towards Abortion Policy: Development of a Heteroskedastic Probit Model of Competing Values." *American Journal of Political Science*, **39**(4), 1055–1082.

Keele LJ, Park DK (2006). *Ambivalent about Ambivalence: A Re-Examination of Heteroskedastic Probit Models.* Unpublished manuscript.

See Also

hetglm

Examples

```
data("AbortionAmbivalence")

## first model for mother's health
ab_health <- hetglm(
  health ~ ethnicity + gender + religion + religiousness + church + erameans + erasupport |
   pros * cons + importance + information + firmness, data = AbortionAmbivalence)
summary(ab_health)

## corresponding model with analytical gradients but numerical Hessian
ab_health2 <- update(ab_health, method = "BFGS", hessian = TRUE)
summary(ab_health2)
```

```
## Alvarez and Brehm (1995), Table 1, p. 1069
## (see also Altman and McDonald, 2003, Supplement, Tables 4-10)
tab1 <- sapply(names(AbortionAmbivalence)[1:7], function(x) {
  f <- as.formula(paste(x,
    "~ ethnicity + gender + religion + religiousness + church + erameans + erasupport",
    "| pros * cons + importance + information + firmness"))
  f0 <- as.formula(paste(x, "~ 1"))
  m <- hetglm(f, data = AbortionAmbivalence)
  m0 <- hetglm(f0, data = model.frame(m))
  c(Percent_yes = as.vector(100 * prop.table(table(AbortionAmbivalence[[x]]))["yes"]),
    coef(m)[c(1:10, 14, 11:13)],
    Heteroscedasticity = as.vector(summary(m)$lrtest[1]),
    N = nobs(m),
    Goodness_of_fit = 2 * as.vector(logLik(m) - logLik(m0))
  )
})
round(tab1, digits = 2)


if(require("AER")) {
## compare Wald tests with different types of standard errors
coeftest(ab_health)
coeftest(ab_health2)
coeftest(ab_health,  vcov = sandwich)
coeftest(ab_health2, vcov = sandwich)
coeftest(ab_health,  vcov = vcovOPG)
coeftest(ab_health2, vcov = vcovOPG)

ab_health_tstat <- cbind(
  "A-Info"     = coeftest(ab_health)[,3],
  "N-Info"     = coeftest(ab_health2)[,3],
  "A-Sandwich" = coeftest(ab_health,  vcov = sandwich)[,3],
  "N-Sandwich" = coeftest(ab_health2, vcov = sandwich)[,3],
  "A-OPG"      = coeftest(ab_health,  vcov = vcovOPG)[,3],
  "N-OPG"      = coeftest(ab_health2, vcov = vcovOPG)[,3]
)
round(ab_health_tstat, digits = 3)
}
```

`BeetleMortality` *Bliss (1935) Beetle Mortality Data*

Description

Mortality of adult flour beetle after five hours' exposure to gaseous carbon disulphide.

Usage

```
data("BeetleMortality")
```

Format

A data frame containing 8 observations on 3 variables.

dose numeric. $\log_{10}$ dose.

died integer. Number killed.

n integer. Number exposed.

Details

The data originates from Bliss (1935) and has been reanalyzed frequently.

Source

Bliss CI (1935). "The Calculation of the Dosage-Mortality Curve." *Annals of Applied Biology*, **22**, 134–167.

References

Aranda-Ordaz F (1981). "On Two Families of Transformations to Additivity for Binary Response Data." *Biometrika*, **68**, 357–363.

Hauck W (1990). "Choice of Scale and Asymmetric Logistic Models." *Biometrical Journal*, **32**, 79–86

Prentice RL (1976). "A Generalization of the Probit and Logit Methods for Dose Response Curves." *Biometrics*, **38**, 761–768.

Pregibon D (1980). "Goodness of Link Tests for Generalized Linear Models." *Journal of the Royal Statistical Society C*, **29**, 15–23.

Examples

```
## data
data("BeetleMortality", package = "glmx")

## various standard binary response models
m <- lapply(c("logit", "probit", "cloglog"), function(type)
  glm(cbind(died, n - died) ~ dose, data = BeetleMortality, family = binomial(link = type)))

## visualization
plot(I(died/n) ~ dose, data = BeetleMortality)
lines(fitted(m[[1]]) ~ dose, data = BeetleMortality, col = 2)
lines(fitted(m[[2]]) ~ dose, data = BeetleMortality, col = 3)
lines(fitted(m[[3]]) ~ dose, data = BeetleMortality, col = 4)
```

glmx *Generalized Linear Models with Extra Parameters*

Description

Estimation of generalized linear models with extra parameters, e.g., parametric links, or families with additional parameters (such as negative binomial).

Usage

```
glmx(formula, data, subset, na.action, weights, offset,
  family = negative.binomial, xlink = "log", control = glmx.control(...),
  model = TRUE, y = TRUE, x = FALSE, ...)

glmx.fit(x, y, weights = NULL, offset = NULL,
  family = negative.binomial, xlink = "log", control = glmx.control())
```

Arguments

formula	symbolic description of the model.
data, subset, na.action	arguments controlling formula processing via model.frame.
weights	optional numeric vector of case weights.
offset	optional numeric vector(s) with an a priori known component to be included in the linear predictor.
family	function that returns a "family" object, i.e., family(x) needs to be a "family" object when x is the numeric vector of extra parameters (by default assumed to be 1-dimensional).
xlink	link object or a character that can be passed to make.link. It should link the extra parameters to real parameters.
control	a list of control arguments as returned by glmx.control.
model, y, x	logicals. If TRUE the corresponding components of the fit (model frame, response, model matrix) are returned. For glmx.fit, x should be a numeric regressor matrix and y should be the response vector.
...	control arguments.

Details

The function glmx is a convenience interface that estimates generalized linear models (GLMs) with extra parameters. Examples would be binary response models with parametric link functions or count regression using a negative binomial family (which has one additional parameter).

Hence, glmx needs a family argument which is a family-generating function depending on one numeric argument for the extra parameters. Then, either profile-likelihood methods can be used for optimizing the extra parameters or all parameters can be optimized jointly.

If the generated family contains a list element loglik.extra for the derivative of the log-likelihood with respect to the extra parameters (i.e., score/gradient contributions), then this is used in the optimization process. This should be a function(y, mu, extra) depending on the observed response y, the estimated mean mu, and the extra parameters.

Value

glmx returns an object of class "glmx", i.e., a list with components as follows. glmx.fit returns an unclassed list with components up to converged.

coefficients	a list with elements "glm" and "extra" containing the coefficients from the respective models,
residuals	a vector of deviance residuals,
fitted.values	a vector of fitted means,
optim	list of optim outputs for maximizing the "profile" and "full" log-likelihood, respectively,
weights	the weights used (if any),
offset	the list of offset vectors used (if any),
n	number of observations,
nobs	number of observations with non-zero weights,
df	number of estimated parameters,
loglik	log-likelihood of the fitted model,
dispersion	estimate of the dispersion parameter (if any),
vcov	covariance matrix of all parameters in the model,
family	a list with elements "glm" and "extra" where the former contains the "family" object at the optimal extra parameters and the latter the family-generating function,
xlink	the link object for the extra parameters,
control	control options used,
converged	logical indicating successful convergence of optim,
call	the original function call,
formula	the formula,
terms	the terms object for the model,
levels	the levels of the categorical regressors,
contrasts	the contrasts corresponding to levels,
model	the full model frame (if model = TRUE),
y	the response vector (if y = TRUE),
x	the model matrix (if x = TRUE).

See Also

glmx.control, hetglm

Examples

```
## artificial data from geometric regression
set.seed(1)
d <- data.frame(x = runif(200, -1, 1))
d$y <- rnbinom(200, mu = exp(0 + 3 * d$x), size = 1)

### negative binomial regression ###

## negative binomial regression via glmx
if(require("MASS")) {
m_nb1 <- glmx(y ~ x, data = d,
  family = negative.binomial, xlink = "log", xstart = 0)
summary(m_nb1)

## negative binomial regression via MASS::glm.nb
m_nb2 <- glm.nb(y ~ x, data = d)
summary(m_nb2)

## comparison
if(require("lmtest")) {
logLik(m_nb1)
logLik(m_nb2)
coeftest(m_nb1)
coeftest(m_nb2)
exp(coef(m_nb1, model = "extra"))
m_nb2$theta
exp(coef(m_nb1, model = "extra")) * sqrt(vcov(m_nb1, model = "extra"))
m_nb2$SE.theta
}}

## if the score (or gradient) contribution of the extra parameters
## is supplied, then estimation can be speeded up:
negbin <- function(theta) {
  fam <- negative.binomial(theta)
  fam$loglik.extra <- function(y, mu, theta) digamma(y + theta) - digamma(theta) +
    log(theta) + 1 - log(mu + theta) - (y + theta)/(mu + theta)
  fam
}
m_nb3 <- glmx(y ~ x, data = d,
  family = negbin, xlink = "log", xstart = 0, profile = FALSE)
all.equal(coef(m_nb1), coef(m_nb3), tolerance = 1e-7)


### censored negative binomial hurdle regression (0 vs. > 0) ###

## negative binomial zero hurdle part via glmx
nbbin <- function(theta) binomial(link = nblogit(theta))
m_hnb1 <- glmx(factor(y > 0) ~ x, data = d,
  family = nbbin, xlink = "log", xstart = 0)
summary(m_hnb1)

## negative binomial hurdle regression via pscl::hurdle
```

```
## (see only zero hurdle part)
if(require("pscl")) {
m_hnb2 <- hurdle(y ~ x, data = d, dist = "negbin", zero.dist = "negbin")
summary(m_hnb2)
}
```

---

glmx.control *Control Parameters for GLMs with Extra Parameters*

Description

Various parameters that control fitting of generalized linear models with extra parameters using glmx.

Usage

```
glmx.control(profile = TRUE, nuisance = FALSE,
  start = NULL, xstart = NULL, hessian = TRUE, method = "BFGS",
  epsilon = 1e-8, maxit = c(500, 25), trace = FALSE,
  reltol = .Machine$double.eps^(1/1.2), ...)
```

Arguments

profile	logical. Should the extra parameters be optimized via profile likelihood (or via the full likelihood of all parameters)?
nuisance	logical. Should the extra parameters be treated as nuisance parameters (i.e., suppressed in subsequent output)?
start	an optional vector with starting values for the GLM coefficients.
xstart	an optional vector with starting values for the extra parameter(s). Must be supplied if there is more than one extra parameter.
hessian	logical or character. Should the hessian be computed to estimate the covariance matrix? If character, hessian can be either "none", "optim" or "numDeriv". The default is the hessian from optim but alternatively hessian from the **numDeriv** package can be used.
method	characters string specifying the method argument passed to optim.
epsilon	numeric convergance tolerance passed to glm.control.
maxit	integer specifying the maxit argument (maximal number of iterations) passed to optim and glm.control. Can also be a vector of length 2.
trace	logical or integer controlling whether tracing information on the progress of the optimization should be produced (passed to optim, and glm.control). Can also be a vector of length 2.
reltol, ...	arguments passed to optim.

Details

All parameters in glmx are estimated by maximum likelihood using optim with control options set in glmx.control. Either the parameters can be found by only optimizing over the extra parameters (and then using glm.fit to estimate the GLM coefficients), or alternatively all parameters can be optimized simultaneously. Covariances are derived numerically using the Hessian matrix returned by optim.

Value

A list with the arguments specified.

See Also

glmx

---

hetglm *Heteroscedastic Binary Response GLMs*

Description

Fit heteroscedastic binary response models via maximum likelihood.

Usage

```
hetglm(formula, data, subset, na.action, weights, offset,
  family = binomial(link = "probit"),
  link.scale = c("log", "sqrt", "identity"),
  control = hetglm.control(...),
  model = TRUE, y = TRUE, x = FALSE, ...)

hetglm.fit(x, y, z = NULL, weights = NULL, offset = NULL,
  family = binomial(), link.scale = "log", control = hetglm.control())
```

Arguments

formula	symbolic description of the model (of type y ~ x or y ~ x | z; for details see below).
data, subset, na.action	arguments controlling formula processing via model.frame.
weights	optional numeric vector of case weights.
offset	optional numeric vector(s) with an a priori known component to be included in the linear predictor(s).
family	family object (including the link function of the mean model).
link.scale	character specification of the link function in the latent scale model.
control	a list of control arguments specified via hetglm.control.

model, y, x	logicals. If TRUE the corresponding components of the fit (model frame, response, model matrix) are returned. For hetglm.fit, x should be a numeric regressor matrix and y should be the numeric response vector (with values in (0,1)).
z	numeric matrix. Regressor matrix for the precision model, defaulting to an intercept only.
...	arguments passed to hetglm.control.

Details

A set of standard extractor functions for fitted model objects is available for objects of class "hetglm", including methods to the generic functions print, summary, coef, vcov, logLik, residuals, predict, terms, update, model.frame, model.matrix, estfun and bread (from the **sandwich** package), and coeftest (from the **lmtest** package).

Value

hetglm returns an object of class "hetglm", i.e., a list with components as follows. hetglm.fit returns an unclassed list with components up to converged.

coefficients	a list with elements "mean" and "scale" containing the coefficients from the respective models,
residuals	a vector of raw residuals (observed - fitted),
fitted.values	a vector of fitted means,
optim	output from the optim call for maximizing the log-likelihood,
method	the method argument passed to the optim call,
control	the control arguments passed to the optim call,
start	the starting values for the parameters passed to the optim call,
weights	the weights used (if any),
offset	the list of offset vectors used (if any),
n	number of observations,
nobs	number of observations with non-zero weights,
df.null	residual degrees of freedom in the homoscedastic null model,
df.residual	residual degrees of freedom in the fitted model,
loglik	log-likelihood of the fitted model,
loglik.null	log-likelihood of the homoscedastic null model,
dispersion	estimate of the dispersion parameter (if any),
vcov	covariance matrix of all parameters in the model,
family	the family object used,
link	a list with elements "mean" and "scale" containing the link objects for the respective models,
converged	logical indicating successful convergence of optim,

call	the original function call,
formula	the original formula,
terms	a list with elements "mean", "scale" and "full" containing the terms objects for the respective models,
levels	a list with elements "mean", "scale" and "full" containing the levels of the categorical regressors,
contrasts	a list with elements "mean" and "scale" containing the contrasts corresponding to levels from the respective models,
model	the full model frame (if model = TRUE),
y	the response vector (if y = TRUE),
x	a list with elements "mean" and "scale" containing the model matrices from the respective models (if x = TRUE).

See Also

Formula

Examples

```
## Generate artifical binary data from a latent
## heteroscedastic normally distributed variable
set.seed(48)
n <- 200
x <- rnorm(n)
ystar <- 1 + x +  rnorm(n, sd = exp(x))
y  <- factor(ystar > 0)

## visualization
par(mfrow = c(1, 2))
plot(ystar ~ x, main = "latent")
abline(h = 0, lty = 2)
plot(y ~ x, main = "observed")

## model fitting of homoscedastic model (m0a/m0b)
## and heteroscedastic model (m)
m0a <- glm(y ~ x, family = binomial(link = "probit"))
m0b <- hetglm(y ~ x | 1)
m <- hetglm(y ~ x)

## coefficient estimates
cbind(heteroscedastic = coef(m),
  homoscedastic = c(coef(m0a), 0))

## summary of correct heteroscedastic model
summary(m)



## Generate artificial binary data with a single binary regressor
```

```
## driving the heteroscedasticity in a model with two regressors
set.seed(48)
n <- 200
x <- rnorm(n)
z <- rnorm(n)
a <- factor(sample(1:2, n, replace = TRUE))
ystar <- 1 + c(0, 1)[a] + x + z + rnorm(n, sd = c(1, 2)[a])
y  <- factor(ystar > 0)

## fit "true" heteroscedastic model
m1 <- hetglm(y ~ a + x + z | a)

## fit interaction model
m2 <- hetglm(y ~ a/(x + z) | 1)

## although not obvious at first sight, the two models are
## nested. m1 is a restricted version of m2 where the following
## holds: a1:x/a2:x == a1:z/a2:z
if(require("lmtest")) lrtest(m1, m2)

## both ratios are == 2 in the data generating process
c(x = coef(m2)[3]/coef(m2)[4], z = coef(m2)[5]/coef(m2)[6])



if(require("AER")) {

## Labor force participation example from Greene
## (5th edition: Table 21.3, p. 682)
## (6th edition: Table 23.4, p. 790)

## data (including transformations)
data("PSID1976", package = "AER")
PSID1976$kids <- with(PSID1976, factor((youngkids + oldkids) > 0,
  levels = c(FALSE, TRUE), labels = c("no", "yes")))
PSID1976$fincome <- PSID1976$fincome/10000

## Standard probit model via glm()
lfp0a <- glm(participation ~ age + I(age^2) + fincome + education + kids,
  data = PSID1976, family = binomial(link = "probit"))

## Standard probit model via hetglm() with constant scale
lfp0b <- hetglm(participation ~ age + I(age^2) + fincome + education + kids | 1,
  data = PSID1976)

## Probit model with varying scale
lfp1 <- hetglm(participation ~ age + I(age^2) + fincome + education + kids | kids + fincome,
  data = PSID1976)

## Likelihood ratio and Wald test
lrtest(lfp0b, lfp1)
waldtest(lfp0b, lfp1)
```

```
## confusion matrices
table(true = PSID1976$participation,
  predicted = fitted(lfp0b) <= 0.5)
table(true = PSID1976$participation,
  predicted = fitted(lfp1) <= 0.5)



## Adapted (and somewhat artificial) example to illustrate that
## certain models with heteroscedastic scale can equivalently
## be interpreted as homoscedastic scale models with interaction
## effects.

## probit model with main effects and heteroscedastic scale in two groups
m <- hetglm(participation ~ kids + fincome | kids, data = PSID1976)

## probit model with interaction effects and homoscedastic scale
p <- glm(participation ~ kids * fincome, data = PSID1976,
   family = binomial(link = "probit"))

## both likelihoods are equivalent
logLik(m)
logLik(p)

## intercept/slope for the kids=="no" group
coef(m)[c(1, 3)]
coef(p)[c(1, 3)]

## intercept/slope for the kids=="yes" group
c(sum(coef(m)[1:2]), coef(m)[3]) / exp(coef(m)[4])
coef(p)[c(1, 3)] + coef(p)[c(2, 4)]

## Wald tests for the heteroscedasticity effect in m and the
## interaction effect in p are very similar
coeftest(m)[4,]
coeftest(p)[4,]

## corresponding likelihood ratio tests are equivalent
## (due to the invariance of the MLE)
m0 <- hetglm(participation ~ kids + fincome | 1, data = PSID1976)
p0 <- glm(participation ~ kids + fincome, data = PSID1976,
  family = binomial(link = "probit"))
lrtest(m0, m)
lrtest(p0, p)

}
```

`hetglm.control` *Control Parameters for Heteroscedastic Binary Response GLMs*

Description

Various parameters that control fitting of heteroscedastic binary response models using hetglm.

Usage

```
hetglm.control(method = "nlminb", maxit = 1000,
  hessian = FALSE, trace = FALSE, start = NULL, ...)
```

Arguments

method	characters string specifying either that nlminb is used for optimization or the method argument passed to optim (typically, "BFGS" or "L-BFGS-B").
maxit	integer specifying the maximal number of iterations in the optimization.
hessian	logical. Should the numerical Hessian matrix from the optim output be used for estimation of the covariance matrix? The default (and only option for nlminb) is to use the analytical expected information rather than the numerical Hessian.
trace	logical or integer controlling whether tracing information on the progress of the optimization should be produced?
start	an optional vector with starting values for all parameters.
...	arguments passed to the optimizer.

Details

All parameters in hetglm are estimated by maximum likelihood using either nlminb (default) or optim with analytical gradients and (by default) analytical expected information. Further control options can be set in hetglm.control, most of which are simply passed on to the corresponding optimizer.

Starting values can be supplied via start or estimated by glm.fit, using the homoscedastic model. Covariances are derived analytically by default. Alternatively, the numerical Hessian matrix returned by optim can be employed, in case this is used for the optimization itself.

Value

A list with the processed specified arguments.

See Also

hetglm

MexicanLabor *Mexican Women's Labor-Force Participation*

Description

Data from the National Survey of Household Income and Expenditures for 1977, Secretaria de Programacion y Presupuesto, Mexico.

Usage

```
data("MexicanLabor")
```

Format

A data frame containing 16 observations on 6 variables.

total integer. Number of women older than 12 years.

laborforce integer. Number of women in labor force.

locality factor with levels `"rural"`/`"urban"`.

age factor with levels `"<= 24"` and `"> 24"` (in years).

income factor with levels `"low"`/`"high"` (household income less or more than $2626.8).

schooling factor with levels `"primary"` (primary school or less) and `"further"` (more than primary school).

Details

The data were first analyzed by Guerrero and Johnson (1982) as an example of a highly asymmetric data set, i.e., the observed proportions are rather low.

Source

Guerrero V, Johnson R (1982). "Use of the Box-Cox Transformation with Binary Response Models." *Biometrika*, **69**, 309–314.

Examples

```
## data
data("MexicanLabor", package = "glmx")

## visualizations
plot(I(laborforce/total) ~ interaction(income, age), data = MexicanLabor)
plot(I(laborforce/total) ~ interaction(schooling, locality), data = MexicanLabor)

## simple logit model
m <- glm(cbind(laborforce, total - laborforce) ~ ., data = MexicanLabor, family = binomial)
m
```

plinks *Parametric Links for Binomial Generalized Linear Models*

Description

Various symmetric and asymmetric parametric links for use as link function for binomial generalized linear models.

Usage

```
gj(phi, verbose = FALSE)
foldexp(phi, verbose = FALSE)
ao1(phi, verbose = FALSE)
ao2(phi, verbose = FALSE)
talpha(alpha, verbose = FALSE, splineinv = TRUE,
  eps = 2 * .Machine$double.eps, maxit = 100)
rocke(shape1, shape2, verbose = FALSE)

gosset(nu, verbose = FALSE)
pregibon(a, b)
nblogit(theta)

angular(verbose = FALSE)
loglog()
```

Arguments

phi, a, b	numeric.
alpha	numeric. Parameter in $[0, 2]$.
shape1, shape2, nu, theta	numeric. Non-negative parameter.
splineinv	logical. Should a (quick and dirty) spline function be used for computing the inverse link function? Alternatively, a more precise but somewhat slower Newton algorithm is used.
eps	numeric. Desired convergence tolerance for Newton algorithm.
maxit	integer. Maximal number of steps for Newton algorithm.
verbose	logical. Should warnings about numerical issues be printed?

Details

Symmetric and asymmetric families parametric link functions are available. Many families contain the logit for some value(s) of their parameter(s).

The symmetric Aranda-Ordaz (1981) transformation

$$y = \frac{2}{\phi} \frac{x^\phi - (1-x)^\phi}{x^\phi + (1-x)^\phi}$$

and the asymmetric Aranda-Ordaz (1981) transformation

$$y = \log([(1-x)^{-\phi} - 1]/\phi)$$

both contain the logit for $\phi = 0$ and $\phi = 1$ respectively, where the latter also includes the complementary log-log for $\phi = 0$.

The Pregibon (1980) two parameter family is the link given by

$$y = \frac{x^{a-b} - 1}{a-b} - \frac{(1-x)^{a+b} - 1}{a+b}.$$

For $a = b = 0$ it is the logit. For $b = 0$ it is symmetric and b controls the skewness; the heavyness of the tails is controlled by a. The implementation uses the generalized lambda distribution `gl`.

The Guerrero-Johnson (1982) family

$$y = \frac{1}{\phi}\left(\left[\frac{x}{1-x}\right]^{\phi} - 1\right)$$

is symmetric and contains the logit for $\phi = 0$.

The Rocke (1993) family of links is, modulo a linear transformation, the cumulative density function of the Beta distribution. If both parameters are set to 0 the logit link is obtained. If both parameters equal 0.5 the Rocke link is, modulo a linear transformation, identical to the angular transformation. Also for `shape1 = shape2` $= 1$, the identity link is obtained. Note that the family can be used as a one and a two parameter family.

The folded exponential family (Piepho, 2003) is symmetric and given by

$$y = \begin{cases} \frac{\exp(\phi x) - \exp(\phi(1-x))}{2\phi} & (\phi \neq 0) \\ x - \frac{1}{2} & (\phi = 0) \end{cases}$$

The t_α family (Doebler, Holling & Boehning, 2011) given by

$$y = \alpha \log(x) - (2-\alpha)\log(1-x)$$

is asymmetric and contains the logit for $\phi = 1$.

The Gosset family of links is given by the inverse of the cumulative distribution function of the t-distribution. The degrees of freedom ν control the heavyness of the tails and is restricted to values > 0. For $\nu = 1$ the Cauchy link is obtained and for $\nu \to \infty$ the link converges to the probit. The implementation builds on `qf` and is reliable for $\nu \geq 0.2$. Liu (2004) reports that the Gosset link approximates the logit well for $\nu = 7$.

Also the (parameterless) angular (arcsine) transformation $y = \arcsin(\sqrt{x})$ is available as a link function.

Value

An object of the class `link-glm`, see the documentation of `make.link`.

References

Aranda-Ordaz F (1981). "On Two Families of Transformations to Additivity for Binary Response Data." *Biometrika*, **68**, 357–363.

Doebler P, Holling H, Boehning D (2012). "A Mixed Model Approach to Meta-Analysis of Diagnostic Studies with Binary Test Outcome." *Psychological Methods*, **17**(3), 418–436.

Guerrero V, Johnson R (1982). "Use of the Box-Cox Transformation with Binary Response Models." *Biometrika*, **69**, 309–314.

Koenker R (2006). "Parametric Links for Binary Response." *R News*, **6**(4), 32–34.

Koenker R, Yoon J (2009). "Parametric Links for Binary Choice Models: A Fisherian-Bayesian Colloquy." *Journal of Econometrics*, **152**, 120–130.

Liu C (2004). "Robit Regression: A Simple Robust Alternative to Logistic and Probit Regression." In Gelman A, Meng X-L (Eds.), *Applied Bayesian Modeling and Causal Inference from Incomplete-Data Perspectives*, Chapter 21, pp. 227–238. John Wiley & Sons.

Piepho H (2003). The Folded Exponential Transformation for Proportions. *Journal of the Royal Statistical Society D*, **52**, 575–589.

Pregibon D (1980). "Goodness of Link Tests for Generalized Linear Models." *Journal of the Royal Statistical Society C*, **29**, 15–23.

Rocke DM (1993). "On the Beta Transformation Family." *Technometrics*, **35**, 73–81.

See Also

`make.link`, `family`, `glmx`, `WECO`

`pregibon` *Pregibon Distribution*

Description

Density, distribution function, quantile function and random generation for the Pregibon distribution with parameters `a` and `b`. It is a special case of the generalized Tukey lambda distribution.

Usage

```
dpregibon(x, a = 0, b = 0, log = FALSE, tol = 1e-12)
ppregibon(q, a = 0, b = 0, lower.tail = TRUE, log.p = FALSE, tol = 1e-12)
qpregibon(p, a = 0, b = 0, lower.tail = TRUE, log.p = FALSE)
rpregibon(n, a = 0, b = 0)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

a, b	distribution parameters.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.
tol	numeric tolerance for computation of the distribution function.

Details

The distribution is a special case of the generalized Tukey lambda distribution and is used by Pregibon (1980) for goodness-of-link testing. See Koenker (2006) and Koenker and Yoon (2009) for more details.

The implementation is based on the corresponding functions for the GeneralisedLambdaDistribution in the **gld** package (King 2013).

The corresponding link generator is available in the function pregibon.

Value

dpregibon gives the probability density function, ppregibon gives the cumulative distribution function, qpregibon gives the quantile function, and rpregibon generates random deviates.

References

King R, Dean B, Klinke S (2016). "Estimation and Use of the Generalised (Tukey) Lambda Distribution." R package version 2.4.1. https://CRAN.R-project.org/package=gld

Koenker R (2006). "Parametric Links for Binary Response." *R News*, **6**(4), 32–34.

Koenker R, Yoon J (2009). "Parametric Links for Binary Choice Models: A Fisherian-Bayesian Colloquy." *Journal of Econometrics*, **152**, 120–130.

Pregibon D (1980). "Goodness of Link Tests for Generalized Linear Models." *Journal of the Royal Statistical Society C*, **29**, 15–23.

See Also

GeneralisedLambdaDistribution, pregibon

Examples

```
## Koenker & Yoon (2009), Figure 2
par(mfrow = c(3, 3))
pregiboncurve <- function(a, b, from, to, n = 301) {
  dp <- function(x) dpregibon(x, a = a, b = b)
  curve(dp, from = from, to = to, n = n,
    xlab = "", ylab = "",
    main = paste("a = ", a, ", b = ", b, sep = ""))
}
pregiboncurve(-0.25, -0.25,  -5, 65)
pregiboncurve(-0.25,  0,    -18, 18)
pregiboncurve(-0.25,  0.25, -65,  5)
pregiboncurve( 0,    -0.25,  -4, 22)
pregiboncurve( 0,     0,     -8,  8)
```

```
pregiboncurve( 0,     0.25, -22,  4)
pregiboncurve( 0.25, -0.25,  -2.4,9)
pregiboncurve( 0.25,  0,     -4,  4)
pregiboncurve( 0.25,  0.25,  -9,  2.4)
par(mfrow = c(1, 1))
```

WECO *Productivity and Quit Behavior of Western Electric Workers*

Description

Partially artificial data about quit behavior of Western Electric workers. (Western Electric was the manufacturing arm of the AT&T corporation during its glory days as a monopolist in the U.S. telephone industry.)

Usage

```
data("WECO")
```

Format

A data frame containing 683 observations on 7 variables.

output productivity in first six months.

sex factor indicating gender.

dex score on a preemployment dexterity exam.

lex years of education.

kwit factor indicating whether the worker quit in the first six months.

tenure duration of employment (see details).

censored logical. Is the duration censored?

Details

The explanatory variables in this example are taken from the study of Klein et al. (1991), but the response variable was altered long ago to improve the didactic impact of the model as a class exercise. To this end, quit dates for each individual were generated according to a log Weibull proportional hazard model.

Source

Online supplements to Koenker (2006) and Koenker and Yoon (2009).

http://www.econ.uiuc.edu/~roger/research/links/links.html

References

Klein R, Spady R, Weiss A (1991). "Factors Affecting the Output and Quit Propensities of Production Workers." *The Review of Economic Studies*, **58**(5), 929–953.

Koenker R (2006). "Parametric Links for Binary Response." *R News*, **6**(4), 32–34.

Koenker R, Yoon J (2009). "Parametric Links for Binary Choice Models: A Fisherian-Bayesian Colloquy." *Journal of Econometrics*, **152**, 120–130.

See Also

plinks

Examples

```
## WECO data
data("WECO", package = "glmx")
f <- kwit ~ sex + dex + poly(lex, 2, raw = TRUE)
## (raw = FALSE would be numerically more stable)

## Gosset model
gossbin <- function(nu) binomial(link = gosset(nu))
m1 <- glmx(f, data = WECO,
  family = gossbin, xstart = 0, xlink = "log")

## Pregibon model
pregibin <- function(shape) binomial(link = pregibon(shape[1], shape[2]))
m2 <- glmx(f, data = WECO,
  family = pregibin, xstart = c(0, 0), xlink = "identity")

## Probit/logit/cauchit models
m3 <- lapply(c("probit", "logit", "cauchit"), function(nam)
  glm(f, data = WECO, family = binomial(link = nam)))

## Probit/cauchit vs. Gosset
if(require("lmtest")) {
lrtest(m3[[1]], m1)
lrtest(m3[[3]], m1)

## Logit vs. Pregibon
lrtest(m3[[2]], m2)
}

## Table 1
tab1 <- sapply(c(m3, list(m1)), function(obj)
  c(head(coef(obj), 5), AIC(obj)))
colnames(tab1) <- c("Probit", "Logit", "Cauchit", "Gosset")
rownames(tab1)[4:6] <- c("lex", "lex^2", "AIC")
tab1 <- round(t(tab1), digits = 3)
tab1

## Figure 4
plot(fitted(m3[[1]]), fitted(m1),
```

```
  xlim = c(0, 1), ylim = c(0, 1),
  xlab = "Estimated Probit Probabilities",
  ylab = "Estimated Gosset Probabilities")
abline(0, 1)
```

Index