

# Package ‘PLreg’

January 23, 2023

**Title** Power Logit Regression for Modeling Bounded Data

**Version** 0.3.1

**Description** Power logit regression models for bounded continuous data, in which the density generator may be normal, Student-t, power exponential, slash, hyperbolic, sinh-normal, or type II logistic. Diagnostic tools associated with the fitted model, such as the residuals, local influence measures, leverage measures, and goodness-of-fit statistics, are implemented. The estimation process follows the maximum likelihood approach and, currently, the package supports two types of estimators: the usual maximum likelihood estimator and the penalized maximum likelihood estimator. More details about power logit regression models are described in Queiroz and Ferrari (2022) <[arXiv:2202.01697](https://arxiv.org/abs/2202.01697)>.

**URL** <https://github.com/ffqueiroz/PLreg>

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.1

**Imports** BBmisc, EnvStats, Formula, gamlss.dist, GeneralizedHyperbolic, methods, nleqslv, stats, VGAM, zipfR

**Suggests** rmarkdown, knitr, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Depends** R (>= 2.10)

**NeedsCompilation** no

**Author** Felipe Queiroz [aut, cre],  
Silvia Ferrari [aut]

**Maintainer** Felipe Queiroz <[ffelipeq@outlook.com](mailto:ffelipeq@outlook.com)>

**Repository** CRAN

**Date/Publication** 2023-01-23 22:00:02 UTC

## R topics documented:

bodyfat_Aeolus . . . . .	2
envelope . . . . .	3
extra.parameter . . . . .	5
Firm . . . . .	6
influence . . . . .	7
methodsPLreg . . . . .	8
PeruVotes . . . . .	9
PL . . . . .	10
plot.PLreg . . . . .	12
PLreg . . . . .	14
PLregcontrol . . . . .	19
residuals.PLreg . . . . .	20
sandwich . . . . .	21

<b>Index</b>	<b>23</b>
--------------	-----------

---

bodyfat_Aeolus	<i>Body Fat of Little Brown Bat</i>
----------------	-------------------------------------

---

### Description

A dataset containing the percent body fat of little brown bats sampled in Aeolus Cave, located in East Dorset, Vermont, in the North Eastern United States.

### Usage

```
data(bodyfat_Aeolus)
```

### Format

A data frame with 159 rows and 4 variables:

**sex** sex of the sampled bat, M for masculine and F for feminine.

**percentfat** percent body fat.

**year** year the bat was sampled.

**days** hibernation time, defined as days since the fall equinox.

### Details

The complete dataset was collected by Cheng et al. (2019) in five different regions of the United States. bodyfat\_Aeolus is the subset of the data collected in Aeolus Cave, located in East Dorset, Vermont, in the North Eastern United States.

### Source

<https://datadryad.org/stash/dataset/doi:10.5061/dryad.sh487nh>

## References

Cheng TL, Gerson A, Moore MS, et al. (2019) Higher fat stores contribute to persistence of little brown bat populations with white-nose syndrome. *J Anim Ecol.* 88:591-600. <https://doi.org/10.1111/1365-2656.12954>

## Examples

```
data("bodyfat_Aeolus")
# Model with zeta = 2
fit <- PLreg(percentfat ~ days + sex + year, data = bodyfat_Aeolus,
             family = "PE", zeta = 2)
summary(fit)
```

---

envelope	<i>Normal Probability Plots with Simulated Envelope of Residuals for PLreg Objects</i>
----------	--

---

## Description

envelope is used to display normal probability plots with simulated envelope of residuals for the power logit models. Currently, three types of residuals are supported: quantile, deviance and standardized residuals.

## Usage

```
envelope(
  object,
  type = c("quantile", "deviance", "standardized"),
  rep = 40,
  conf = 0.95,
  xlab,
  ylab,
  main,
  envcol,
  ylim,
  xlim
)
```

## Arguments

object	fitted model object of class "PLreg".
type	character specifying the type of residuals to be used, see <a href="#">residuals.PLreg</a> . Default is type = "standardized".
rep	a positive integer representing the number of iterations to calculate the simulated envelopes. Default is rep=40.
conf	a numeric value in the interval (0,1) that represents the confidence level of the simulated envelopes. Default is conf=0.95.

xlab	character specifying the label for $x$ axis (optional). Default is "Quantile $N(0, 1)$ ".
ylab	character specifying the label for $y$ axis (optional). Default is the name of the used residual.
main	character specifying the overall title for the plot.
envcol	character specifying the color of the envelope.
ylim, xlim	numeric values, specifying the left/lower limit and the right/upper limit of the scale.

### Details

The envelope uses the idea of Atkinson (1985) to create normal probability plots with simulated envelope. Under the correct model, approximately  $100 \cdot \text{conf}$  of the residuals are expected to be inside the envelope.

### Value

envelope returns normal probability plot with simulated envelopes for the residuals.

### References

Queiroz, F. F. and Ferrari, S. L. P. (2022). Power logit regression for modeling bounded data. *arXiv:2202.01697*.

Atkinson, A. C. (1985) Plots, transformations and regression: an introduction to graphical methods of diagnostic regression analysis. *Oxford Science Publications*, Oxford.

### See Also

[PLreg](#), [residuals.PLreg](#)

### Examples

```
data("Firm")

fitPL <- PLreg(firmcost ~ sizelog + indcost | sizelog + indcost, data = Firm,
              family = "SLASH", zeta = 2.13)
summary(fitPL)

envelope(fitPL, type = "standardized")
envelope(fitPL, type = "quantile")
envelope(fitPL, type = "deviance")
```

---

extra.parameter      *Procedure to Select the Extra Parameter for PLreg Objects*

---

**Description**

The `extra.parameter` function is used to select the extra parameter of some power logit models. It provides plots of  $-2\log\text{Lik}$  and the Upsilon measure (see Queiroz and Ferrari (2021)) versus  $\zeta$ , the extra parameter.

**Usage**

```
extra.parameter(object, lower, upper, grid = 10, graph = TRUE)
```

**Arguments**

object	fitted model object of class "PLreg".
lower	a numeric value representing the lower limit of the interval for the extra parameter.
upper	a numeric value representing the upper limit of the interval for the extra parameter.
grid	a positive integer representing the number of points in the plots. Default is <code>grid=10</code> . If <code>grid</code> is less than 10, then <code>grid=10</code> .
graph	logical. If <code>graph = TRUE</code> the plots are shown, if <code>graph = FALSE</code> the plots are not shown. Default is <code>graph = TRUE</code> .

**Value**

`extra.parameter` returns a list with five objects:

<code>zeta.Ups</code>	The selected zeta based on the Upsilon measure.
<code>zeta.loglik</code>	The selected zeta based on $-2\log\text{Lik}$ .
<code>zeta.values</code>	The values of zeta used in the graphs.
<code>Upsilon.values</code>	$-2\log\text{Lik}$ evaluated at each value of zeta.
<code>loglik.values</code>	Upsilon measure evaluated at each value of zeta.

**References**

Queiroz, F. F. and Ferrari, S. L. P. (2022). Power logit regression for modeling bounded data. *arXiv:2202.01697*.

**See Also**

[PLreg](#)

**Examples**

```

data("bodyfat_Aeolus")

#Initial model with zeta = 2
fit1 <- PLreg(percentfat ~ days + sex + year, data = bodyfat_Aeolus,
              family = "PE", zeta = 2)
summary(fit1)
# Choosing the best value for zeta

extra.parameter(fit1, lower = 1, upper = 4, grid = 15)

```

---

Firm

*Firm Cost*


---

**Description**

A dataset on risk management practices of 73 firms.

**Usage**

```
data(Firm)
```

**Format**

A data frame with 73 rows and 7 variables:

**firmcost** total property and casualty premiums and uninsured losses as a percentage of total assets.

**assume** per occurrence retention amount as a percentage of total assets.

**cap** indicates that the firm owns a captive insurance company; 1 if the firm uses a captive, 0 otherwise.

**size $\log$**  logarithm of total assets.

**indcost** a measure of the firm's industry risk.

**central** a measure of the importance of the local managers in choosing the amount of risk to be retained.

**soph** a measure of the degree of importance in using analytical tools.

**Details**

The dataset was introduced and analyzed by Schmit and Roth (1990) and is available in the personal web page of Professor E. Frees (Wisconsin School of Business Research). The response variable is FIRM COST, smaller values of firm cost are attributed to firms that have a good risk management performance.

**Source**

<https://instruction.bus.wisc.edu/jfrees/jfreesbooks/Regression%20Modeling/BookWebDec2010/CSVData/RiskSurvey.csv>

## References

Schmit, J.T. and Roth, K. (1990). Cost effectiveness of risk management practices. *Journal of Risk and Insurance*. 57:455-470.

## Examples

```
data("Firm")
fitPL <- PLreg(firmcost ~ sizelog + indcost | sizelog + indcost,
              data = Firm,
              family = "SLASH",
              zeta = 2.13)
summary(fitPL)
plot(fitPL, type = "standardized")
```

---

influence

*Influence Diagnostics for PLreg Objects*

---

## Description

The influence function provides two influence measures and the generalized leverage for power logit regression models.

## Usage

```
influence(model, graph = TRUE, ...)
```

## Arguments

model	fitted model object of class "PLreg".
graph	logical. If graph = TRUE the plots are shown, if graph = FALSE the plots are not shown. Default is graph = TRUE.
...	currently not used.

## Value

influence returns a list with three objects:

case.weights	The values of $h_{max}$ eigenvector based on case weights perturbation scheme (see Queiroz and Ferrari (2021)).
totalLI	The total local influence (see Lesaffre and Verbeke (1998))
GL	The diagonal elements of the generalized leverage matrix.

## References

Queiroz, F. F. and Ferrari, S. L. P. (2022). Power logit regression for modeling bounded data. *arXiv:2202.01697*.

**See Also**

[PLreg](#), [residuals.PLreg](#), [envelope](#), [plot.PLreg](#)

**Examples**

```
data("Firm")

fitPL <- PLreg(firmcost ~ sizelog + indcost | sizelog + indcost,
              data = Firm, family = "SLASH", zeta = 2.13)

influence_measures = influence(fitPL, graph = FALSE)
plot(influence_measures$case.weights, type = "h", ylim = c(0,1))
plot(influence_measures$totalLI, type = "h", ylim = c(0,6))
plot(Firm$sizelog, influence_measures$GL, pch = "+")
```

---

methodsPLreg

*Methods for PLreg Objects*

---

**Description**

Some S3 Methods for PLreg regression models.

**Usage**

```
## S3 method for class 'PLreg'
summary(object, type = "standardized", ...)

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

## S3 method for class 'summary.PLreg'
print(x, ...)

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

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

## S3 method for class 'PLreg'
logLik(object, ...)

## S3 method for class 'PLreg'
model.matrix(object, model = c("median", "dispersion"), ...)
```



## Arguments

object, x	fitted model object of class "PLreg".
type	character specifying the type of residuals to be included in the summary output, see <a href="#">residuals.PLreg</a> .
...	currently not used.
model	character specifying for which component of the model the coefficients/covariance are extracted.

## Details

A set of methods for objects of class "PLreg", including methods for the functions [summary](#) and [vcov](#), which print the estimated coefficients along with some other information and presents the covariance matrix, respectively. The [summary](#) also presents the partial Wald tests for the model parameters. Finally, [summary](#) returns an object of class "summary.PLreg" containing information to be printed using the `print` method.

## See Also

[PLreg](#)

## Examples

```
data("bodyfat_Aeolus")

fitPL <- PLreg(percentfat ~ 1, data = bodyfat_Aeolus,
              family = "SN", zeta = 1.6)

fitPL
summary(fitPL)
coef(fitPL, model = "median")
vcov(fitPL)
logLik(fitPL)
AIC(fitPL)
```

---

PeruVotes

*Peru Blank Votes*

---

## Description

A dataset on the blank votes in the 2006 Peruvian general election.

## Usage

```
data(PeruVotes)
```

**Format**

A data frame with 194 rows and 2 variables:

**votes** proportion of blank votes in the 2006 Peruvian general election.

**HDI** Human Development Index.

**Details**

The dataset was collected by Bayes et al. (2012) and the response variable is votes, proportion of blank votes in the 2006 Peruvian general election. Bayes et al. (2012) analyzed the influence of the Human Development Index (HDI) on the proportion of blank votes using a beta rectangular regression model. Lemonte and Bazan (2015) also analyzed the data using GJS regression models.

**Source**

<https://www.undp.org/es/peru>

**References**

Bayes, C., Bazan, J. L. and Garcia, C. (2012). A new robust regression model for proportions. *Bayesian Analysis*. 7:771-796

Lemonte, A. J. and Bazan, J. L. (2015). New class of Johnson SB distributions and its associated regression model for rates and proportions. *Biometrical Journal*. 58:727-746.

**Examples**

```
data("PeruVotes")
fitPL <- PLreg(votes ~ HDI | HDI,
              data = PeruVotes,
              family = "TF",
              zeta = 5,
              control = PLreg.control(lambda = 1))
summary(fitPL)
plot(fitPL, type = "standardized")
```

**Description**

Density, distribution function, quantile function and random generation for power logit distributions.

**Usage**

```
dPL(x, mu, sigma, lambda, zeta = 2, family, log = FALSE)
```

```
pPL(q, mu, sigma, lambda, zeta = 2, family, lower.tail = TRUE, log.p = FALSE)
```

```
qPL(p, mu, sigma, lambda, zeta = 2, family, lower.tail = TRUE, log.p = FALSE)
```

```
rPL(n, mu, sigma, lambda, zeta = 2, family)
```

**Arguments**

x, q	vector of quantiles.
mu	vector of medians.
sigma	vector of dispersion parameters.
lambda	vector of skewness parameters.
zeta	vector of extra parameters.
family	string that specifies the family used to define the power logit distribution. The family is NO, TF, LO, PE, SHN, Hyp and SLASH for normal, Student-t, type II logistic, power exponential, sinh-normal, hyperbolic and slash distribution, respectively.
log, log.p	logical; if TRUE, probabilities p are given as log(p). Default is FALSE.
lower.tail	logical; if TRUE (default), probabilities are $P(X \leq x)$ otherwise, $P(X > x)$ .
p	vector of probabilities.
n	number of observations. If $\text{length}(n) > 1$ , the length is taken to be the number required.

**Details**

If zeta is not specified, it assumes the default value 2.

The power logit distribution has density

$$f(y; \mu, \sigma, \lambda) = \lambda r(z^2) / (\sigma y (1 - y^\lambda)),$$

for  $y \in (0, 1)$ , in which  $z = [\text{logit}(y^\lambda) - \text{logit}(\mu^\lambda)] / \sigma$ ,  $r(\cdot)$  is the density generator and  $\mu \in (0, 1)$ ,  $\sigma > 0$  and  $\lambda > 0$  are the median, dispersion and skewness of the distribution.

It is possible to consider  $\lambda = 0$  to obtain the limiting case, the log-log distribution. This distribution has density

$$f(y; \mu, \sigma, \lambda) = r(z^2) / (\sigma y (-\log(y))),$$

for  $y \in (0, 1)$ , in which  $z = [-\log(-\log(y)) - (-\log(-\log(\mu)))] / \sigma$ .

The family argument defines the density generator  $r(\cdot)$ , which may depend on an extra parameter (zeta).

**Value**

dPL gives the density, pPL gives the distribution function, qPL gives the quantile function, and rPL generates random variables.

## References

Queiroz, F. F. and Ferrari, S. L. P. (2022). Power logit regression for modeling bounded data. *arXiv preprint arXiv:2202.01697*.

## Examples

```
dPL(0.2, mu = 0.3, sigma = 1, lambda=1, zeta = 2, family = "PE")
mu = 0.3; sigma = 1; lambda = 2
set.seed(1)
PLsample = rPL(1000, mu, sigma, lambda, family = "SN", zeta = 2.5)
hist(PLsample, prob = TRUE, breaks = 15, main = "", las = 1)
curve(dPL(x, mu, sigma, lambda, family = "SN", zeta = 2.5),
      from = 0.01, to = 0.8, add = TRUE, col = "red")
rug(PLsample)

x = seq(0.01, 0.9, 0.01)
y = dPL(x, mu, sigma, lambda, family = "Hyp", zeta = 2)
plot(x, y, type = "l", lwd = 2, las = 1)

x1 = seq(0.01, 0.4, 0.01)
y1 = dPL(x1, mu, sigma, lambda, family = "Hyp", zeta = 2)
polygon(c(x1, 0.4, 0), c(y1, 0, 0), col = "lightblue")
text(mu-0.025, 1, paste("P(Y<0.4) = ",
                      round(pPL(0.4, mu, sigma, lambda,
                              family = "Hyp",
                              zeta = 2), 2)),
      font = 1, cex = 0.8)

plot(x, pPL(x, mu, sigma, lambda, family = "PE", zeta = 1.3),
     type = "l", las = 1, lwd = 2,
     ylab = expression(P(Y<y)),
     xlab = "y")
p = pPL(0.5, mu, sigma, lambda, family = "PE", zeta = 1.3)
q = qPL(p, mu, sigma, lambda, family = "PE", zeta = 1.3)
points(q, p, pch = 16, col = 2, cex = 1.5)
text(0.55, 0.83, paste("(", 0.5, ", ", round(p, 2), ")"), font = 2,
     cex = 0.8, col = "red")
```

---

plot.PLreg

*Diagnostic Plots for PLreg Objects*

---

## Description

This function provides plots for diagnostic analysis of power logit regression models.

## Usage

```
## S3 method for class 'PLreg'
plot(x, which = 1:4, type = "standardized", pch = "+", las = 1, cex = 0.8, ...)
```

**Arguments**

x	fitted model object of class "PLreg".
which	numeric specifying a subset of plots (numbers between 1 and 7). Default is which = 1:4.
type	character specifying the type of residuals, see <a href="#">residuals.PLreg</a> . Default is type = "standardized".
pch, las, cex, ...	graphical parameters (see <a href="#">par</a> )

**Details**

The plot method for [PLreg](#) objects provides 7 types of diagnostic plots in the following order.

**Residuals vs indexes of obs.** An index plot of the residuals versus indexes of observations.

**Case-weight perturbation** An index plot of local influence based on the case-weight perturbation scheme.

**Generalized leverage** A dispersion diagram of the generalized leverage versus the predicted values.

**Residuals vs linear predictor** A dispersion diagram of the residuals versus the linear predictors.

**Normal probability plot** A normal probability plot of the residuals.

**Predicted vs observed values** A dispersion diagram of the predicted values versus the observed values.

**Residuals vs  $v(z)$  function** A dispersion diagram of the  $v(z)$  function versus the residuals. For some power logit models, the  $v(z)$  function may be interpreted as weights in the estimation process. If family = "NO", the  $v(z)$  function is constant.

The which argument can be used to select a subset of the implemented plots. Default is which = 1:4.

**Value**

plot method for [PLreg](#) objects returns 7 types of diagnostic plots.

**See Also**

[PLreg](#), [residuals.PLreg](#), [envelope](#), [influence](#)

**Examples**

```
data("Firm")

fitPL <- PLreg(firmcost ~ sizelog + indcost | sizelog + indcost, data = Firm,
              family = "SLASH", zeta = 2.13)
par(mfrow = c(3,3))
plot(fitPL, type = "standardized")
par(mfrow = c(1, 1))
```

**Description**

PLreg is used to fit power logit regression model for continuous and bounded variables via maximum likelihood approach. Both median and dispersion of the response variable are modeled through parametric functions.

**Usage**

```
PLreg(  
  formula,  
  data,  
  subset,  
  na.action,  
  family = c("NO", "LO", "TF", "PE", "SN", "SLASH", "Hyp"),  
  zeta = NULL,  
  link = c("logit", "probit", "cloglog", "cauchit", "loglog"),  
  link.sigma = NULL,  
  type = c("pML", "ML"),  
  control = PLreg.control(...),  
  model = TRUE,  
  y = TRUE,  
  x = FALSE,  
  ...  
)  
  
PLreg.fit(  
  X,  
  y,  
  S = NULL,  
  family,  
  type = "pML",  
  zeta = zeta,  
  link = "logit",  
  link.sigma = "log",  
  control = PLreg.control()  
)
```

**Arguments**

`formula` a symbolic description of the model. See details for further information.  
`data`, `subset`, `na.action` arguments controlling formula processing via `model.frame`.

family	a description of the symmetric distribution to be used for generating the power logit model. Supported families include "NO", "LO", "TF", "PE", "Hyp", "SHN" and "SLASH", which correspond to the power logit normal, type II logistic, Student-t, power exponential, hyperbolic, sinh-normal, and slash distributions, respectively.
zeta	a numeric value or numeric vector that represents the extra parameter of the distribution. For the PL-NO and PL-LO models, no extra parameter is needed.
link	an optional character that specifies the link function of the median submodel ( $\mu$ ). The "logit", "probit", "cloglog", "cauchit", "loglog" functions are supported. The logit function is the default.
link.sigma	an optional character that specifies the link function of the dispersion submodel ( $\sigma$ ). The "log", "sqrt" functions are supported. The default is log.
type	character specifying the type of estimator for the skewness parameter. Currently, penalized maximum likelihood ("pML") and maximum likelihood ("ML") are supported. If the skewness parameter is fixed, ML type is used.
control	a list of control arguments specified via <a href="#">PLreg.control</a> .
model, y, x	logicals. If TRUE the corresponding components of the fit (model frame, response, model matrix) are returned. For <a href="#">PLreg.fit</a> , y must be the numeric response vector (with values in (0,1)).
...	arguments passed to <a href="#">PLreg.control</a> .
X	numeric regressor matrix for the median submodel.
S	numeric regressor matrix for the dispersion submodel.

## Details

The power logit regression models, proposed by Queiroz and Ferrari (2021), is useful in situations when the response variable is continuous and bounded on the unit interval (0, 1). The median and the dispersion parameters are modeled through parametric link functions. The models depend on a skewness parameter (called  $\lambda$ ). When the skewness parameter is fixed and equal to 1, the power logit models coincide with the GJS regression models (Lemonte and Bazan, 2016). Queiroz and Ferrari (2021) suggest using a penalized maximum likelihood method to estimate the parameters. This method is implemented in PLreg by default when  $\lambda$  is not fixed. If convergence is not reached, maximum likelihood estimation is performed. The estimation process uses [optim](#). If no starting values are specified, the PLreg function uses those suggested by Queiroz and Ferrari (2021). This function also fits the log-log regression models by setting  $\lambda$  at zero ( $\lambda = 0$  represents  $\lambda \rightarrow 0^+$ ).

The formulation of the model has the same structure as in the usual functions [lm](#) and [glm](#). The argument `formula` could comprise of three parts (separated by the symbols " " and "|"), namely: observed response variable in the unit interval, predictor of the median submodel, with link function `link` and predictor of the dispersion submodel, with `link.sigma` link function. If the model has constant dispersion, the third part may be omitted and the link function for sigma is "log" by default. The skewness parameter `lambda` may be treated as fixed or not (default). If `lambda` is fixed, its value must be specified in the `control` argument.

Some methods are available for objects of class "PLreg", see [plot.PLreg](#), [summary.PLreg](#), [coef.PLreg](#), [vcov.PLreg](#), and [residuals.PLreg](#), for details and other methods.

**Value**

PLreg returns an object of class "PLreg" with the following components (the PLreg.fit returns elements up to v).

coefficients	a list with the "median", "dispersion" and "skewness" (if lambda = NULL) coefficients.
residuals	a vector of the raw residuals (the difference between the observed and the fitted response).
fitted.values	a vector with the fitted values of the median submodel.
optim	a list with the output from optim. When lambda is not fixed, if type = "pML", the output refers to the iterative process of the median and dispersion parameters only and, if type = "ML", on the maximization of the likelihood for all the parameters.
family	a character specifying the family used.
method	the method argument passed to the optim call.
control	the control arguments passed to the optim call.
start	a vector with the starting values used in the iterative process.
nobs	number of observations.
df.null	residual degrees of freedom in the null model (constant median and dispersion), i.e., $n - 3$ .
df.residual	residual degrees of freedom in the fitted model.
lambda	value of the skewness parameter lambda (NULL when lambda is not fixed).
loglik	log-likelihood of the fitted model.
vcov	covariance matrix of all the parameters.
pseudo.r.squared	pseudo R-squared value.
Upsilon.zeta	an overall goodness-of-fit measure.
link	a list with elements "median" and "dispersion" containing the link objects for the respective models.
converged	logical indicating successful convergence of the iterative process.
zeta	a numeric specifying the value of zeta used in the estimation process.
type	a character specifying the estimation method used.
v	a vector with the v(z) values for all the observations (see Queiroz and Ferrari(2021)).
call	the original function call.
formula	the formula used.
terms	a list with elements "median", "dispersion" and "full" containing the term objects for the respective models.
levels	a list with elements "median", "dispersion" and "full" containing the levels of the categorical regressors.



contrasts	a list with elements "median" and "dispersion" containing the contrasts corresponding to levels from the respective models.
model	the full model frame (if y = TRUE).
y	the response variable (if y = TRUE).
x	a list with elements "median" and "dispersion" with the matrices from the median and dispersion submodels (if x = TRUE).

**Author(s)**

Francisco Felipe de Queiroz (<ffelipeq@outlook.com>) and Silvia L. P. Ferrari.

**References**

Queiroz, F. F. and Ferrari, S. L. P. (2022). Power logit regression for modeling bounded data. *arXiv:2202.01697*.

Lemonte, A. J. and Bazan, J. L. (2015). New class of Johnson SB distributions and its associated regression model for rates and proportions. *Biometrical Journal*. 58:727-746.

**See Also**

[summary.PLreg](#), [PLreg.control](#), [residuals.PLreg](#)

**Examples**

```
#### Body fat data
data("bodyfat_Aeolus")

#Initial model with zeta = 2
fit1 <- PLreg(percentfat ~ days + sex + year, data = bodyfat_Aeolus,
              family = "PE", zeta = 2)
summary(fit1)
# Choosing the best value for zeta
# extra.parameter(fit1, lower = 1, upper = 4, grid = 15)

# Using zeta = 1.7
fit2 <- PLreg(percentfat ~ days + sex + year, data = bodyfat_Aeolus,
              family = "PE", zeta = 1.7)
summary(fit2)

# Fixing lambda = 1
fit3 <- PLreg(percentfat ~ days + sex + year, data = bodyfat_Aeolus,
              family = "PE", zeta = 1.7,
              control = PLreg.control(lambda = 1))
summary(fit3)

# Comparing the AIC and Upsilon values between fit2 and fit3
AIC(fit2) < AIC(fit3) # TRUE
fit2$Upsilon.zeta < fit3$Upsilon.zeta #TRUE

#### Firm cost data
```

```

data("Firm")

fitPL <- PLreg(firmcost ~ sizelog + indcost | sizelog + indcost,
              data = Firm,
              family = "SLASH",
              zeta = 2.13)
summary(fitPL)
#extra.parameter(fitPL, lower = 1.2, upper = 4, grid = 10)
#plot(fitPL, type = "standardized")
#envelope(fitPL, type = "standardized")

fitPL_wo72 <- PLreg(firmcost ~ sizelog + indcost | sizelog + indcost,
                   data = Firm[-72,],
                   family = "SLASH",
                   zeta = 2.13)
fitPL_wo15 <- PLreg(firmcost ~ sizelog + indcost | sizelog + indcost,
                   data = Firm[-15,],
                   family = "SLASH",
                   zeta = 2.13)
fitPL_wo16 <- PLreg(firmcost ~ sizelog + indcost | sizelog + indcost,
                   data = Firm[-16,],
                   family = "SLASH",
                   zeta = 2.13)

coef.mu      <- coef(fitPL)[1:3]
coef.mu_wo72 <- coef(fitPL_wo72)[1:3]
coef.mu_wo15 <- coef(fitPL_wo15)[1:3]
coef.mu_wo16 <- coef(fitPL_wo16)[1:3]

plot(Firm$indcost, Firm$firmcost,
     pch = "+",
     xlab = "indcost",
     ylab = "firmcost")
#identify(Firm$indcost, Firm$firmcost)
covariate = matrix(c(rep.int(1, 1000),
                      rep(median(Firm$sizelog), 1000),
                      seq(0, 1.22, length.out = 1000)),
                   ncol = 3)
lines(covariate[,3],
      as.vector(fitPL$link$median$linkinv(covariate**coef.mu)),
      type = "l")
lines(covariate[,3],
      as.vector(fitPL$link$median$linkinv(covariate**coef.mu_wo72)),
      type = "l", lty = 2, col = "blue")
lines(covariate[,3],
      as.vector(fitPL$link$median$linkinv(covariate**coef.mu_wo15)),
      type = "l", lty = 3, col = "red")
lines(covariate[,3],
      as.vector(fitPL$link$median$linkinv(covariate**coef.mu_wo16)),
      type = "l", lty = 4, col = "green")
parameters = c("pML",
               "pML w/o 72",
               "pML w/o 15",

```

```

                                "pML w/o 16")
legend(x = 0.5,
       y = 0.8,
       legend = parameters,
       col = c("black", "blue", "red", "green"),
       lty = c(1, 2, 3, 4),
       cex = 0.6)

```

---

 PLregcontrol

*Auxiliary for Controlling PL Fitting*


---

### Description

Parameters that control fitting of power logit regression models using [PLreg](#).

### Usage

```

PLreg.control(
  lambda = NULL,
  method = "BFGS",
  maxit = 2000,
  trace = FALSE,
  start = NULL,
  ...
)

```

### Arguments

lambda	numeric indicating the value of the skewness parameter lambda (if NULL, lambda will be estimated).
method	character specifying the method argument passed to <a href="#">optim</a> .
maxit, trace, ...	arguments passed to <a href="#">optim</a>
start	an optional vector with starting values for median and dispersion submodels (starting value for lambda must not be included).

### Details

The `PLreg.control` controls the fitting process of power logit models. Almost all the arguments are passed directly to [optim](#), which is used to estimate the parameters. Starting values for median and dispersion submodels may be supplied via `start`. If the estimation process is to be performed with a fixed skewness parameter, a value must be specified in `lambda`. If `lambda = 0`, a log-log regression model will be estimated.

### Value

A list with components named as the arguments.

**See Also**[PLreg](#)**Examples**

```
data("PeruVotes")

fitPL <- PLreg(votes ~ HDI | HDI, data = PeruVotes,
              family = "TF", zeta = 5, control = PLreg.control(lambda = 1))
summary(fitPL)
```

---

`residuals.PLreg`*Residuals Method for PLreg Objects*

---

**Description**

The function provides three types of residuals for power logit regression models: quantile, deviance and standardized.

**Usage**

```
## S3 method for class 'PLreg'
residuals(object, type = c("quantile", "deviance", "standardized"), ...)
```

**Arguments**

<code>object</code>	fitted model object of class "PLreg".
<code>type</code>	character specifying the type of residuals to be used.
<code>...</code>	currently not used.

**Details**

The *quantile residuals* is based on Dunn and Smyth (1996) idea. The residuals are well-defined for all the distributions in the power logit class and have, approximately, a standard normal distribution in large samples if the model is correctly specified.

The *deviance residuals* are based on the log-likelihood contributions of each observation in the sample. The distribution of this residual is unknown (both exact and asymptotic), except for the power logit normal model, which is, approximately, standard normal.

The *standardized residuals* are a standardized form of the ordinary residual. These residuals take into account the diagonal elements of the  $H$  matrix, being useful for detecting leverage observations. The distribution of the standardized residuals is unknown.

**Value**

`residuals` method for object of class "PLreg" returns a vector with the residuals of the type specified in the `type` argument.

**References**

Queiroz, F. F. and Ferrari, S. L. P. (2022). Power logit regression for modeling bounded data. *arXiv:2202.01697*.

Dunn, P. K. and Smyth, G. K. (1996) Randomized quantile residuals. *Journal of Computational and Graphical Statistics*, 5:236-244.

**See Also**

[PLreg](#), [plot.PLreg](#), [envelope](#), [influence](#)

**Examples**

```
data("PeruVotes")
fitPL <- PLreg(votes ~ HDI | HDI, data = PeruVotes,
              family = "TF", zeta = 5)

res_quantile = residuals(fitPL, type = "quantile")
res_standardized = residuals(fitPL, type = "standardized")

plot(res_standardized, pch = "+", ylim = c(-6, 6))
abline(h = -3, lty = 2)
abline(h = 3, lty = 2)

qqnorm(res_quantile)
qqline(res_quantile)
```

---

sandwich

*Sandwich Variance and Covariance Matrix for PLreg Objects*

---

**Description**

The `sandwich` function provides an estimate for the asymptotic variance and covariance matrix of the parameter estimators of the power logit (or log-log) regression models based on the sandwich estimator described below.

**Usage**

```
sandwich(object)
```

**Arguments**

`object` fitted model object of class "PLreg".

**Value**

`extra.parameter` returns a matrix containing the sandwich variance and covariance matrix estimate.

**References**

Queiroz, F. F. and Ferrari, S. L. P. (2022). Power logit regression for modeling bounded data. *arXiv:2202.01697*.

**See Also**

[PLreg](#)

**Examples**

```
data("Firm")

fit <- PLreg(percentfat ~ days + sex + year, data = bodyfat_Aeolus,
             family = "PE", zeta = 2)
sandwich(fit)
```

# Index

- \* **datasets**
  - bodyfat\_Aeolus, 2
  - Firm, 6
  - PeruVotes, 9
- bodyfat\_Aeolus, 2
- coef.PLreg, 15
- coef.PLreg (methodsPLreg), 8
- dPL (PL), 10
- envelope, 3, 8, 13, 21
- extra.parameter, 5
- Firm, 6
- glm, 15
- influence, 7, 13, 21
- lm, 15
- logLik.PLreg (methodsPLreg), 8
- methodsPLreg, 8
- model.frame, 14
- model.matrix.PLreg (methodsPLreg), 8
- optim, 15, 19
- par, 13
- PeruVotes, 9
- PL, 10
- plot.PLreg, 8, 12, 15, 21
- PLreg, 4, 5, 8, 9, 13, 14, 19–22
- PLreg.control, 15, 17
- PLreg.control (PLregcontrol), 19
- PLreg.fit, 15
- PLregcontrol, 19
- pPL (PL), 10
- print.PLreg (methodsPLreg), 8
- print.summary.PLreg (methodsPLreg), 8
- qPL (PL), 10
- residuals.PLreg, 3, 4, 8, 9, 13, 15, 17, 20
- rPL (PL), 10
- sandwich, 21
- summary, 9
- summary.PLreg, 15, 17
- summary.PLreg (methodsPLreg), 8
- vcov, 9
- vcov.PLreg, 15
- vcov.PLreg (methodsPLreg), 8