PLreg: An R Package for Modeling Bounded Continuous Data

by Francisco F. Queiroz and Silvia L.P. Ferrari

Abstract The power logit class of distributions is useful for modeling continuous data on the unit interval, such as fractions and proportions. It is very flexible and the parameters represent the median, dispersion and skewness of the distribution. Based on the power logit class, Queiroz and Ferrari (2023b, *Statistical Modelling*) proposed the power logit regression models. The dependent variable is assumed to have a distribution in the power logit class, with its median and dispersion linked to regressors through linear predictors with unknown coefficients. We present the R package **PLreg** which implements a suite of functions for working with power logit class of distributions and the associated regression models. This paper describes and illustrates the methods and algorithms implemented in the package, including tools for parameter estimation, diagnosis of fitted models, and various helper functions for working with power logit distributions, including density, cumulative distribution, quantile, and random number generating functions. Additional examples are presented to show the ability of the **PLreg** package to fit generalized Johnson SB, log-log, and inflated power logit regression models.

1 Introduction

Continuous proportion data frequently appear in areas including medicine, biology, and economics. Some concrete examples are vegetation cover fraction, mortality rate, and body fat percentage. Frequently, the interest lies in predicting or explaining the behaviour of proportions from a set of other variables. A natural approach is to use a regression model in which the response variable takes values on the unit interval. The most frequently employed model for bounded data is the beta regression model (Ferrari and Cribari-Neto, 2004) and its extensions. Other models used for bounded continuous data include, for example, rectangular beta (Bayes et al., 2012), simplex (Barndorff-Nielsen and Jørgensen, 1991; Zhang and Qiu, 2014), log-Lindley (Gómez-Déniz et al., 2014), CDF-quantile (Smithson and Shou, 2017), generalized Johnson SB (GJS; Lemonte and Bazán (2016)). Some of these models are implemented in R. For instance, the beta, simplex, and CDF-quantile regression models can be fitted using the packages **betareg** (Zeileis et al., 2021), **simplexreg** (Zhang et al., 2016), and **cdfquantreg** (Shou and Smithson, 2022), respectively. The **gamlss** (Stasinopoulos and Rigby, 2007) package can also be used to fit beta and simplex regression models.

Recently, Queiroz and Ferrari (2023b) proposed a new class of regression models useful for modeling continuous data with bounded support. The models employ a new class of distributions called power logit (PL), indexed by the median, dispersion and skewness parameters. The PL distributions are constructed from standard symmetric distributions assigned to the power logit transformation of the variable that has support on $(0, 1)$. The power logit transformation is defined in Queiroz and Ferrari (2023b) as $t(y; \lambda) = \log[y^\lambda/(1-y^\lambda)]$, for $\lambda > 0$ and $y \in (0, 1)$; it reduces to the logit transformation when $\lambda = 1$. The PL distributions may also depend on an extra parameter that indexes the underlying symmetric distribution; for example, the degrees-of-freedom parameter of the Student-t distribution. The extra parameter adds extra flexibility, which can be used, for example, to deal with outliers. The PL distributions are more flexible than two-parameter distributions, such as the beta, simplex, and CDF-quantile distributions. The class of PL regression models has the GJS regression models as a particular case ($\lambda = 1$), with the advantage that the skewness parameter λ provides extra flexibility to fit highly skewed data. Applications in real data presented in Queiroz and Ferrari (2023b) reveal that the PL regression models are helpful for modeling continuous proportions.

The new R package **PLreg** provides a broad set of tools for fitting PL regression models and performing diagnostic analysis. The package is implemented in R and available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=PLreg`. This paper describes and illustrates the methods and algorithms implemented in the package. It also presents some examples to demonstrate the ability of the package to fit generalized Johnson SB, log-log, and inflated power logit regression models.

The remaining of this paper is organized as follows. Section 2.2 presents the PL distributions and the associated regression models. Section 2.3 describes the implementation of the PL distributions and the PL regression models in the **PLreg** package. Section 2.4 gives detailed illustrations of the use of the **PLreg** package for modeling continuous bounded data in different scenarios. The paper closes with a brief discussion outlining the features of the **PLreg** package.

2 Power logit regression models

2.1 Power logit distributions

The PL distributions (Queiroz and Ferrari, 2023b) are defined from a transformation of a continuous random variable whose distribution is standard symmetric with probability density function $r(z^2)$, $z \in \mathbb{R}$, in which $r(z) > 0$, for $z \geq 0$, with $\int_0^\infty z^{-1/2} r(z) dz = 1$. The function $r(\cdot)$ is called the density generator function. Let Y be a continuous random variable with support $(0, 1)$ and let:

$$Z = h(Y; \mu, \sigma, \lambda) = \frac{1}{\sigma}\left[\log\left(\frac{Y^\lambda}{1-Y^\lambda}\right) - \log\left(\frac{\mu^\lambda}{1-\mu^\lambda}\right)\right],$$

where $0 < \mu < 1$, $\sigma > 0$, and $\lambda > 0$. If Z has a standard symmetric distribution with density generator function $r(\cdot)$, we say that Y has a PL distribution with parameters μ, σ and λ, and density generator function $r(\cdot)$. We write $Y \sim \text{PL}(\mu, \sigma, \lambda; r)$. The density generator function $r(\cdot)$ may depend on an extra parameter, denoted here by ζ. The distribution of Y depends on the distribution chosen for Z. For instance, if Z has a standard normal distribution, then Y has a PL normal distribution; if Z has a standard Student-t distribution with ζ degrees-of-freedom, then Y has a PL Student-t distribution with extra parameter ζ.

The probability density function (pdf) of $Y \sim \text{PL}(\mu, \sigma, \lambda; r)$ is:

$$f_Y(y; \mu, \sigma, \lambda) = \frac{\lambda}{\sigma y(1-y^\lambda)} r(z^2), \quad y \in (0, 1),$$

where $z = h(y; \mu, \sigma, \lambda)$. The cumulative distribution function (cdf) of Y is $F_Y(y; \mu, \sigma, \lambda) = R(z)$, where $R(\cdot)$ is the cdf of Z.

The GJS class of distributions is a particular case of the PL distributions when $\lambda = 1$. Other particular cases are the logit normal distribution (Johnson, 1949), the L-Logistic distribution (da Paz et al., 2019), and the logit slash distribution (Korkmaz, 2020), obtained by taking $\lambda = 1$ and Z as a standard normal, type II logistic, and slash random variable, respectively.

The PL distributions have some interesting properties. For instance, the parameters μ, σ and λ represent the median, dispersion and skewness of the distributions, and they have as a limiting case when $\lambda \to 0^+$, the class of log-log distributions, defined in Queiroz and Ferrari (2023b).

2.2 Power logit regression models

The PL regression models are defined as follows. Let $Y_1, \ldots, Y_n$ be n independent random variables, where $Y_i \sim \text{PL}(\mu_i, \sigma_i, \lambda; r)$, for $i = 1, \ldots, n$, and

$$\begin{aligned} d_1(\mu_i) &= \boldsymbol{x_i}^\top \boldsymbol{\beta} = \eta_{1i}, \\ d_2(\sigma_i) &= \boldsymbol{s_i}^\top \boldsymbol{\tau} = \eta_{2i}, \end{aligned} \tag{1}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top \in \mathbb{R}^p$, $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_q)^\top \in \mathbb{R}^q$ and $\lambda > 0$ are the unknown parameters, which are assumed to be functionally independent and $p + q + 1 < n$; η_{1i} and η_{2i} are the linear predictors; $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^\top$ and $\boldsymbol{s}_i = (s_{i1}, \ldots, s_{iq})^\top$ are the covariates. We assume that $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\top$ and $\mathbf{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n]^\top$ have column rank p and q, respectively. In addition, we assume that the link functions $d_1 : (0, 1) \to \mathbb{R}$ and $d_2 : (0, \infty) \to \mathbb{R}$ are strictly monotonic and twice differentiable. Some examples of link functions for the median submodel are: $d_1(\mu) = \log\{\mu/(1-\mu)\}$ (logit); $d_1(\mu) = \Phi^{-1}(\mu)$ (probit), where $\Phi^{-1}(\cdot)$ is the cdf of a standard normal random variable; $d_1(\mu) = -\log\{-\log\mu\}$ (log-log); and $d_1(\mu) = \log\{-\log(1-\mu)\}$ (complementary log-log). For the dispersion submodel, the log link, $d_2(\sigma) = \log\sigma$, is the natural choice.

The log-log regression models are a limiting case of the PL regression models when $\lambda \to 0^+$. The GJS regression models (Lemonte and Bazán, 2016) are obtained by taking $\lambda = 1$.

The estimation of $\boldsymbol{\theta} = (\boldsymbol{\beta}^\top, \boldsymbol{\tau}^\top, \lambda)^\top$ is based on the maximum likelihood approach. The log-likelihood function of $\boldsymbol{\theta}$ for the observed sample $y_1, \ldots, y_n$ is:

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ell_i(\mu_i, \sigma_i, \lambda),$$

where $\ell_i = \ell_i(\mu_i, \sigma_i, \lambda) = \log\lambda - \log\sigma_i - \log\{1 - y_i^\lambda\} + \log\{r(z_i^2)\} + c$, $z_i = h(y_i; \mu_i, \sigma_i, \lambda)$, and c does not depend on $\boldsymbol{\theta}$. The maximum likelihood estimate (mle) of $\boldsymbol{\theta}$, denoted by $\widehat{\boldsymbol{\theta}}$, can be obtained by

solving simultaneously the nonlinear system of equations $\mathbf{U}(\boldsymbol{\theta}) = \mathbf{0}_{p+q+1}$, which does not have a closed form, where $\mathbf{U}(\boldsymbol{\theta})$ is the score function and $\mathbf{0}_{p+q+1}$ denotes a $(p+q+1)$-dimensional vector of zeros. Queiroz and Ferrari (2023b) also proposed a penalized maximum likelihood estimator (pmle), which is recommended when the sample size is small. The pmle, denoted by $\widetilde{\boldsymbol{\theta}}$, is computed through numerical optimization as follows.

i. Compute $\widetilde{\lambda}$ such that:
$$\widetilde{\lambda} = \underset{\lambda>0}{\operatorname{argmax}}\, \ell_p^*(\lambda),$$
where $\ell_p^*(\lambda)$ is the penalized profile log-likelihood for λ; see Queiroz and Ferrari (2023b, Equation 7).

ii. Compute $\widetilde{\boldsymbol{\beta}}$ and $\widetilde{\boldsymbol{\tau}}$ by maximizing $\ell(\boldsymbol{\beta}, \boldsymbol{\tau}, \widetilde{\lambda})$.

The extra parameter ζ, if any, is selected by minimizing the overall goodness-of-fit measure Y_ζ, defined as:
$$\mathrm{Y}_\zeta = n^{-1} \sum_{i=1}^{n} |\Phi^{-1}[R(\widetilde{z}^{(i)})] - v^{(i)}|,$$
where $\widetilde{z}^{(i)}$ is the ith order statistic of $\widetilde{z}$, $v^{(i)}$ is the mean of the ith order statistic in a random sample of size n of the standard normal distribution and $\Phi(\cdot)$ is the cdf of the standard normal distribution. Alternatively, ζ may be selected by maximizing $\ell(\widetilde{\boldsymbol{\theta}})$.

Some diagnostic tools for the PL regression models are presented in Queiroz and Ferrari (2023b), including quantile, deviance, and standardized residuals, local influence methods, and a generalized leverage measure. Applications and further details on inference methods are found in Queiroz and Ferrari (2023b).

3 R implementation

The **PLreg** package allows fitting the PL regression models. The package is organized in a similar way to other packages for fitting regression models, such as **betareg** and **simplexreg**. The estimation process is based on the likelihood theory, and two estimators are available: the mle and the pmle. Diagnostic tools for evaluating the fitted model are also implemented. Currently, the package includes methods for computing three types of residuals: quantile, deviance, and standardized residuals. Local influence measures, leverage measures, and goodness-of-fit statistics are also available. Additionally, the package supports PL regression models with the skewness parameter λ fixed, i.e., the package also allows fitting GJS and log-log regression models.

3.1 Power logit distributions in the PLreg package

Currently, the **PLreg** package includes seven distributions of the PL class: the PL normal, PL Student-t, PL power exponential, PL slash, PL hyperbolic, PL sinh-normal, and PL type II logistic distributions. **PLreg** provides the `dPL()`, `pPL()`, and `qPL()` functions to compute the probability density function, cumulative distribution function and quantile function of the PL distributions. Also, the `rPL()` function may be used to generate random samples of variables with a PL distribution. The basic usages of these functions are:

```
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)
```

The main arguments for these functions are `mu`, `sigma`, `lambda` and `family`, specifying the parameters μ, σ, and λ and the corresponding density generator function $r(\cdot)$, that is, the distribution of the symmetric distribution Z. If `lambda = 0`, those functions provide results for the log-log distributions. If the density generator function depends on an extra parameter, its value must be specified in the `zeta` argument. On the other hand, if it does not depend on an extra parameter, the argument `zeta` is ignored. The arguments `x` and `q` are the vector of quantiles, `p` is a vector of probabilities, and `n` is the number of random numbers to be generated. Other arguments are `log`, `log.p` and `lower.tail`. If `log = TRUE`, then the logarithm of the probability density function will be returned. If `log.p = TRUE`, then the logarithm of the cumulative distribution function will be returned and the quantile function will

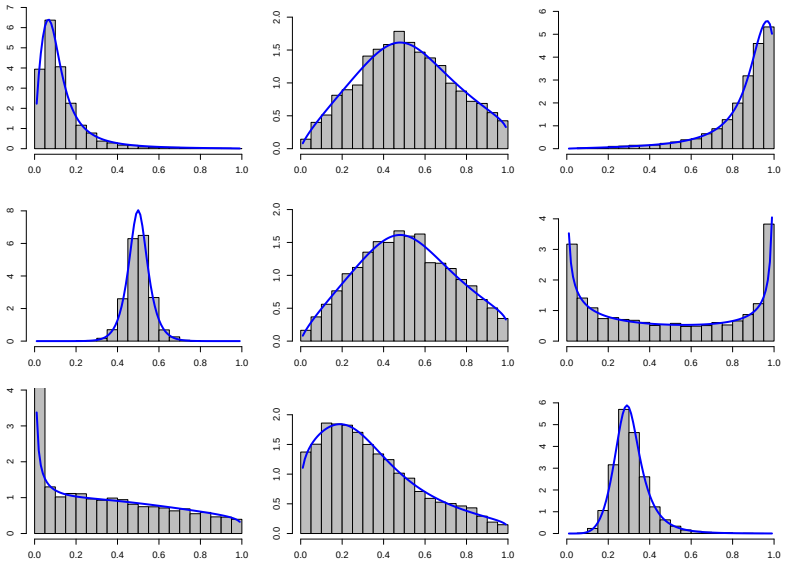

Figure 1: Histograms of random numbers of the PL hyperbolic distributions with $\zeta = 1.2$ with different values for μ, σ, and λ. First line: $\mu = (0.1, 0.5, 0.9)$, $\sigma = 1$, $\lambda = 1.5$; second line: $\mu = 0.5$, $\sigma = (0.2, 1, 3)$, $\lambda = 1.5$; third line: $\mu = 0.3$, $\sigma = 1$, $\lambda = (0.01, 1, 5)$. Solid lines are corresponding to the respective PL hyperbolic density.

be computed for $\exp(p)$. If `lower.tail = FALSE`, then one minus the cumulative distribution function will be returned and the quantile function will be computed for $1 - p$.

In the following, we present the density generator function of all PL distributions implemented in the **PLreg** package:

- PL normal (`family = "NO"`): $r(z) = (2\pi)^{-1/2} \exp(-z/2)$;
- PL Student-t (`family = "TF"`): $r(z) = \zeta^{\zeta/2} B(1/2, \zeta/2)^{-1} (\zeta + z)^{-(\zeta+1)/2}$, $\zeta > 0$ and $B(\cdot, \cdot)$ is the beta function;
- PL type II logistic (`family = "LO"`): $r(z) = \exp\{-z^{1/2}\}(1 + \exp\{-z^{1/2}\})^{-2}$;
- PL power exponential (`family = "PE"`): $r(z) = \zeta / [p(\zeta) 2^{1+1/\zeta} \Gamma(1/\zeta)] \times \exp\left\{-z^{\zeta/2} / (2p(\zeta)^{\zeta})\right\}$, $\zeta > 0$ and $p(\zeta)^2 = 2^{-2/\zeta} \Gamma(1/\zeta) / \Gamma(3/\zeta)$;
- PL slash (`family = "SLASH"`): $r(z) = (\zeta/\sqrt{2\pi})\,(z/2)^{-(\zeta+1/2)}\, G\,(\zeta + 1/2, z/2)$, for $z > 0$, and $r(z) = 2\zeta / [(2\zeta + 1)\sqrt{2\pi}]$, for $z = 0$, where $\zeta > 0$ and $G(a, x) = \int_0^x t^{a-1} e^{-t} dt$ is the lower incomplete gamma function. When $\zeta = 1$ the slash distribution coincides with the canonical slash distribution;
- PL hyperbolic (`family = "Hyp"`): $r(z) = \exp\left\{-\zeta\sqrt{1+z}\right\} / (2\zeta K_1(\zeta))$, with $K_s(\zeta) = \int_0^{\infty} \frac{x^{s-1}}{2} \times \exp\{-\frac{\zeta}{2}\left(x + \frac{1}{x}\right)\} \mathrm{d}x$, is the modified Bessel function of third-order and index s.
- PL sinh-normal (`family = "SN"`): $r(z) = 1/(\zeta\sqrt{2\pi}) \cosh(z^{1/2}) \exp\left[-2/\zeta^2 \sinh^2(z^{1/2})\right]$, where $\zeta > 0$ and $\sinh(\cdot)$ and $\cosh(\cdot)$ represent the hyperbolic sine and cosine functions, respectively.

Figure 1 illustrates the use of the `rPL()` and `dPL()` functions showing the distribution of random numbers generated from PL hyperbolic distributions with $\zeta = 1.2$.

3.2 Power logit regression models

The main model-fitting function of the **PLreg** package is `PLreg()`, which is similar to the other functions for implementing regression models in R. The basic usage of the `PLreg()` function is:

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

The argument formula may comprise three parts (separated by the symbols " " and "|"), namely: the observed response variable with values on $(0,1)$, the linear predictor of the median submodel and the linear predictor of the dispersion submodel (for further details about the `formula` argument, see Zeileis and Croissant (2010)). For instance, `formula = y  x1 + x2 + x3 | z1 + z2` describes y for the response variable, x_1, x_2, and x_3 for the median submodel, and z_1 and z_2 for the dispersion submodel. The model is fitted with constant dispersion if the third part of the argument `formula` is omitted. So, a PL regression model with constant dispersion may be specified either by `formula = y  x1 + x2 + x3` or `formula = y  x1 + x2 + x3 | 1`. The available link functions for the median submodel are `"logit"`, `"probit"`, `"cloglog"`, `"cauchit"`, and `"loglog"`. For the dispersion submodel, two link functions are allowed: `"log"` and `"sqrt"`. The default link functions are `"logit"` for the median submodel and `"log"` for the dispersion submodel. There are two other important arguments: `family` and `zeta`. The argument `family` specifies the symmetric distribution used for generating the PL model; the currentlu supported families are `"NO"`, `"LO"`, `"TF"`, `"PE"`, `"Hyp"`, `"SN"`, and `"SLASH"`. For the `"TF"`, `"PE"`, `"Hyp"`, `"SN"`, and `"SLASH"` families the extra parameter must be specified in the `zeta` argument.

The estimation process is carried out via `optim()` with control options set in `PLreg.control()`. It is based on the maximum likelihood method. Currently, two estimators are supported: the usual maximum likelihood estimator (`"ML"`) and a penalized maximum likelihood estimator (`"pML"`); this should be specified in the `type` argument. If the skewness parameter (λ) is fixed, only the usual maximum likelihood estimator is supported. In this case, a value should be specified in the control argument through the `PLreg.control()` function. For instance, `control = PLreg.control(lambda = 1)` and `control = PLreg.control(lambda = 0)` lead to the GJS and the log-log regression models, respectively; note that $\lambda = 0$ represents $\lambda \to 0^+$. Also, if `type = "ML"`, `optim()` uses analytical gradients in the iterative process; if `type = "pML"`, analytical gradients are used only in the iterative process to estimate the parameters of the median and dispersion submodels. By default, the starting values are chosen as described in Queiroz and Ferrari (2023b), but they may be user-supplied through the `PLreg.control()` function.

Once the model has been fitted, an object of `S3` class '`PLreg`' is produced. A list of some of the components of this object is presented in Table 1. The complete list can be obtained in the reference manual of the package (Queiroz and Ferrari, 2023a). Several methods are available for objects of class '`PLreg`'. The `summary()` method presents a standard output, with coefficient estimates, standard errors, partial Wald statistics and p values for the regression coefficients, as well as the overall goodness-of-fit measure (Y_ζ), the pseudo R^2, and other metrics. The argument `type` in `summary()` specifies the type of residuals included in the output: `"standardized"`, `"quantile"` or `"deviance"`. The `plot()` method draws graphs for diagnostic and influence analyses. Table 2 presents a list with all the available functions and methods.

The `extra.parameter()` function can be used to select the extra parameter of some PL models. The basic use is as follows:

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

This function provides a graph of `-2loglik` and Y_ζ as functions of ζ, the extra parameter. The `object` argument is an object of class '`PLreg`'; `lower` and `upper` are the lower and upper limits of the interval for the extra parameter, respectively; and `grid` is the number of values of the extra parameter for which the measures are evaluated.

4 Examples using the PLreg package

We present some examples to illustrate the features of the **PLreg** package. We use a simulated dataset and three datasets available in the package: `bodyfat_Aeolus`, `Firm`, and `PeruVotes`. These analyses were conducted using R version 4.2.2.

4.1 `bodyfat_Aeolus` data: IID setting

For a simple illustration of the **PLreg** package, we consider the `bodyfat_Aeolus` data reported in Cheng et al. (2019). The dataset used here has 159 observations and was collected in Aeolus Cave, located in East Dorset, Vermont, in the USA. The bats were sampled during the winter of 2009 (covering the winter season from October 2008 to April 2009) and 2016 (October 2015 to April 2016). Here, the interest lies in modeling the proportion of body fat of little brown bats (`percentfat`) using the PL distributions. The data can be loaded by:

```
R> data("bodyfat_Aeolus", package = "PLreg")
```

Component	Description
`coefficients`	list of the fitted model coefficients.
`residuals`	vector of raw residuals.
`fitted.values`	vector of the fitted values (fitted median for each observation).
`optim`	list with the `optim()` output. For unfixed λ, if `type = "pML"`, the output is based on the iterative process for estimating $\boldsymbol{\beta}$ and γ; and, if `type = "ML"`, it is based on the iterative process for estimating the whole parameter vector.
`family`	character specifying the underlying symmetric distribution.
`method`	optimization method used in `optim()`. Default is `"BFGS"`.
`control`	control arguments passed to `optim()`.
`start`	vector with the starting values used to initialize the optimization process.
`nobs`	number of observations.
`df.null`	residual degrees of freedom in the null model (constant median and dispersion).
`df.residual`	residual degrees of freedom in the fitted model.
`lambda`	value of the skewness parameter λ (`NULL` when λ is not fixed).
`loglik`	log-likelihood of the fitted model.
`loglikp`	penalized profile log-likelihood for λ.
`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 value indicating whether the optimization converged successfully.
`zeta`	a numeric specifying the value of ζ 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 (2023b) for details.

Table 1: List of the components of an object of the 'PLreg' class.

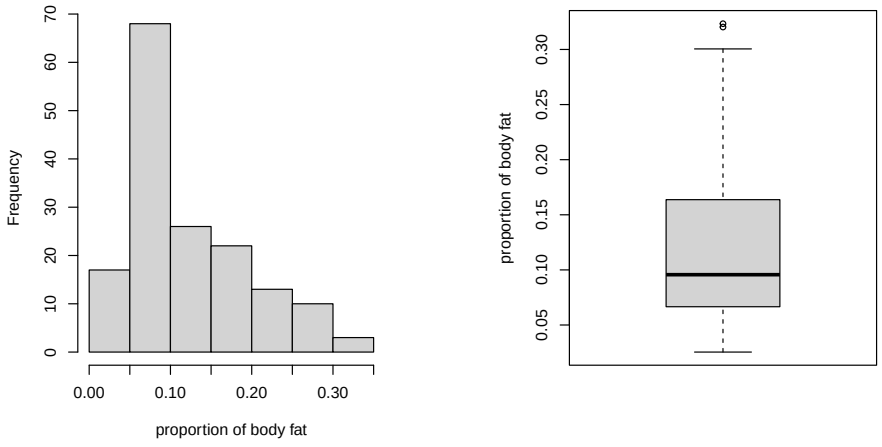

Figure 2: Histogram (left side) and boxplot (right side) of the response variable – `bodyfat_Aeolus` data.

and the histogram and boxplot of the response variable are presented in Figure 2. Some summary measures of `percentfat` are presented below. Note that the distributions of this variable is right-skewed and have some values close to zero, with range 2.5% − 32%.

Function	Description
`print()`	prints the coefficients estimates.
`summary()`	output for the fitted model. Returns an object of class `"summary.PLreg"` containing the relevant information about the fit and has a `print()` method.
`coef()`	extracts the coefficients of the fitted model.
`vcov()`	variance and covariance matrix.
`logLik()`	extracts the fitted log-likelihood function.
`model.matrix()`	extracts model matrix of model components.
`AIC()`	computes information criteria (AIC, BIC, ...).
`residuals()`	extracts residuals for the fitted model (quantile, standardized and deviance). Default is the standardized residual.
`plot()`	presents some diagnostic plots. Currently, seven types of plots are available: index plot of residuals, local influence plot based on the case-weight perturbation scheme, scatter plot of the generalized leverage versus the predicted values, scatter plot of the residuals versus the linear predictors, normal probability plot of the residuals, scatter plot of the predicted values versus the observed values, and a scatter plot of the $v(z)$ function versus the residuals (for some PL models, $v(z)$ may be interpreted as weights in the estimation process).
`influence()`	provides two influence measures and the generalized leverage for PL regression models.
`envelope()`	returns a normal probability plot with simulated envelopes for the residuals.
`extra.parameter()`	provides plots for selecting the extra parameter, if any.
`CI.lambda()`	provides plot of the profile (penalized) likelihood ratio statistics for λ. Used to obtain confidence intervals for λ.
`sandwich()`	provides an estimate for the asymptotic variance and covariance matrix of the parameter estimators of the PL regression models based on the sandwich estimator.

Table 2: List with the methods and functions of an object of the 'PLreg' class.

```
R> summary(bodyfat_Aeolus$percentfat)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.02544 0.06658 0.09565 0.12168 0.16371 0.32336
```

We now fit the `percentfat` variable using the PL normal and PL sinh-normal distributions. For the PL sinh-normal distribution, we first fit the distribution with a fixed value of ζ, e.g., $\zeta = 1$, and then we use the `extra.parameter()` function to select an optimal value for ζ. In the **PLreg** package, it can be done via:

```
R> PLNO <- PLreg(percentfat ~ 1, data = bodyfat_Aeolus, family = "NO")
R> PLSN.aux <- PLreg(percentfat ~ 1, data = bodyfat_Aeolus, family = "SN",
+                    zeta = 1)
R> extra.parameter(PLSN.aux, lower = 1, upper = 4, grid = 10)

Estimates for zeta are:
zeta.Ups = 1.67
zeta.loglik = 2

> PLSN <- PLreg(percentfat ~ 1, data = bodyfat_Aeolus, family = "SN",
+               zeta = 1.67)
```

The `extra.parameter()` returns the optimal values for ζ based on two measures and plots these measures as functions of ζ; see Figure 3. We choose `zeta = 1.67`.

To select between the PL normal and PL sinh-normal distributions, we compute the Y_ζ and the AIC for both fits:

```
R> PL_NO <- round(c(PLNO$Upsilon.zeta, AIC(PLNO)), 3)
R> PL_SN <- round(c(PLSN$Upsilon.zeta, AIC(PLSN)), 3)
```

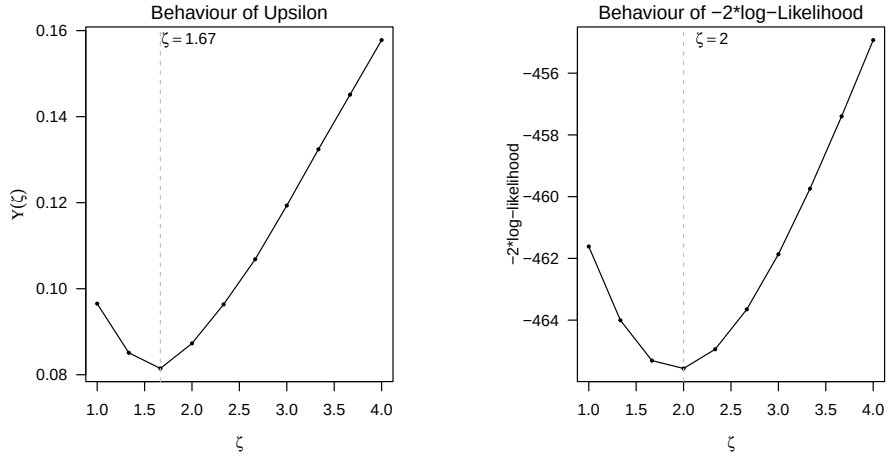

Figure 3: Plot returned by the `extra.parameter()` for selecting an optimal value for ζ of the fit of the PL sinh-normal distribution – `bodyfat_Aeolus` data.

```
R> measures <- rbind(PL_NO, PL_SN)
R> colnames(measures) <- c("Upsilon", "AIC")
R> measures

      Upsilon      AIC
PL_NO   0.114 -443.771
PL_SN   0.082 -453.942
```

Since the values of Y_ζ and AIC for the PL sinh-normal fit are smaller than those of the PL normal fit, we select the PL sinh-normal distribution. The `summary` output of the PL sinh-normal fit is presented in the following:

```
R> summary(PLSN)

Call:
PLreg(formula = percentfat ~ 1, data = bodyfat_Aeolus, family = "SN",
    zeta = 1.67)

Standardized residuals:
    Min      1Q  Median      3Q     Max
-2.4654 -0.8318 -0.2065  0.7453  2.0551

Coefficients (median model with logit link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.11477    0.05416  -39.05   <2e-16 ***

Sigma coefficients (dispersion model with log link):
        Estimate Std. Error z value Pr(>|z|)
(sigma)   0.2107     0.4506   0.468     0.64

Lambda coefficient:
         Estimate Std. Error
(lambda)    1.438      0.764
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Family: PL - SN ( 1.67 ) (Power logit sinh-normal)
Estimation method: pML (penalized maximum likelihood)
Log-likelihood:   230 on 3 Df
Upsilon statistic: 0.08151
AIC: -453.9
Number of iterations in BFGS optimization: 9
```

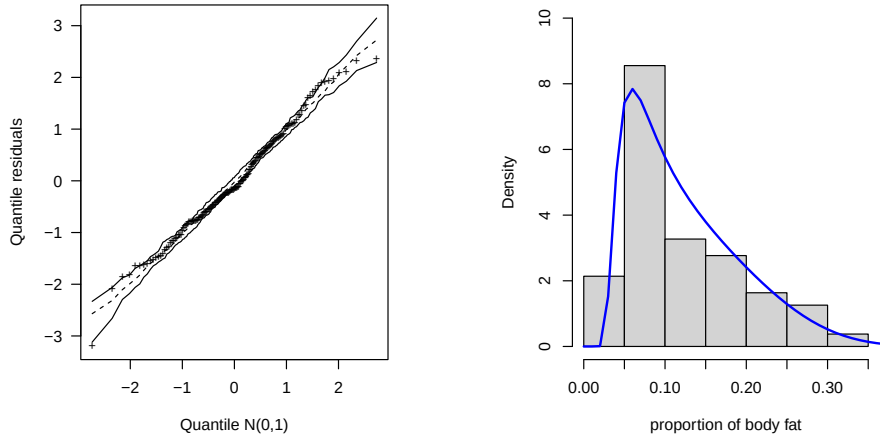

Figure 4: Normal probability plot of the quantile residual with simulated envelope for the `PLSN` fit and the histogram of `percentfat` with the estimated density – `bodyfat_Aeolus` data.

The estimated median of the body fat proportion based on the fit is $\exp(-2.11477)/[1+\exp(-2.11477)] \approx 10.8\%$, close to the sample median. Figure 4 presents the normal probability plot of the quantile residual with simulated envelope for the `PLSN` fit and the histogram of `percentfat` with the estimated density. These plots can be obtained as follows:

```
R> set.seed(180123)
R> envelope(PLSN)
R> hist(bodyfat_Aeolus$percentfat, main = " ",
+      xlab = "proportion of body fat", prob = TRUE, ylim = c(0, 10))
R> curve(dPL(x, PLSN$fitted.values[1], exp(PLSN$coefficients$dispersion),
+           PLSN$coefficients$skewness, zeta = 1.67, family = "SN"), 0 , 1,
+        add = TRUE, lwd = 2, col= "blue")
```

Using the delta method, an approximated 95% confidence interval for the median of the body fat proportion is:

$$\left[\widetilde{\mu} \mp 1.96 \times \text{se}(\widetilde{\beta}) \frac{\exp(\widetilde{\beta})}{[1+\exp(\widetilde{\beta})]^2}\right] = [0.097,\ 0.118].$$

4.2 `Firm` data: PL regression model

We now use the `Firm` data to replicate the application presented in Queiroz and Ferrari (2023b, Section 6.2). The dataset was introduced by Schmit and Roth (1990) and presents information on the risk management practices of 73 firms. The response variable is `firmcost`, defined as premiums plus uninsured losses as a percentage of the total assets. It is a measure of the firm's risk management cost-effectiveness. Queiroz and Ferrari (2023b) start the analysis with the PL slash regression model with varying dispersion, employing two covariates: `sizelog`, the logarithm of total assets, and `indcost`, a measure of the firm's industry risk. This model can be fitted via:

```
R> data("Firm", package = "PLreg")

R> Firm_slash2 <- PLreg(firmcost ~ indcost + sizelog | indcost + sizelog,
+                       data = Firm, family = "SLASH", zeta = 2)
R> extra.parameter(Firm_slash2, lower = 1, upper = 2.5, grid = 30)

Estimates for zeta are:
zeta.Ups = 1.88
zeta.loglik = 1.93

R> Firm_slash <- PLreg(firmcost ~ indcost + sizelog | indcost + sizelog,
+                   data = Firm, family = "SLASH", zeta = 1.88)
R> summary(Firm_slash)
```

```
Call:
PLreg(formula = firmcost ~ indcost + sizelog | indcost + sizelog,
        data = Firm, family = "SLASH", zeta = 1.88)

Standardized residuals:
    Min      1Q  Median      3Q     Max
-2.1220 -0.6253  0.0251  0.6548  4.6194

Coefficients (median model with logit link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   3.8223     1.0196   3.749 0.000178 ***
indcost       2.3117     0.8062   2.867 0.004140 **
sizelog      -0.9082     0.1225  -7.416 1.21e-13 ***

Sigma coefficients (dispersion model with log link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.56915    0.78874  -0.722    0.471
indcost      0.36623    0.54062   0.677    0.498
sizelog      0.07455    0.09979   0.747    0.455

Lambda coefficient:
         Estimate Std. Error
(lambda)    2.035      1.196
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Family: PL - SLASH ( 1.88 ) (Power logit slash)
Estimation method: pML (penalized maximum likelihood)
Log-likelihood:   123 on 7 Df
Pseudo R-squared: 0.4177
Upsilon statistic: 0.06723
AIC: -232.1
Number of iterations in BFGS optimization: 10
```

The standard errors presented in the `summary()` output is computed from the observed information matrix. Queiroz and Ferrari (2023b) employ the sandwich matrix to obtain standard errors. Standard errors are computed from the sandwich matrix by using the `sandwich()` function in the **PLreg** package as follows:

```
R> sand.matrix <- sandwich(Firm_slash)
R> se <- sqrt(diag(sand.matrix))
R> se

       (Intercept)             indcost             sizelog (sigma)_(Intercept)
        1.30397074          1.05026082          0.16419123          0.54569178
   (sigma)_indcost     (sigma)_sizelog            (lambda)
        0.59534023          0.08767362          0.91164845
```

All the covariates are statistically significant for the median submodel but not for the dispersion submodel. Then, the authors fit the PL slash regression model with constant dispersion and select $\zeta = 2.29$. The model can be fitted as follows:

```
R> Firm_slash.CD <- PLreg(firmcost ~ indcost + sizelog,
+                         data = Firm, family = "SLASH", zeta = 2.29)
R> summary(Firm_slash.CD)

Call:
PLreg(formula = firmcost ~ indcost + sizelog,
        data = Firm, family = "SLASH", zeta = 2.29)

Standardized residuals:
    Min      1Q  Median      3Q     Max
-2.1133 -0.6590  0.0546  0.7168  5.9131

Coefficients (median model with logit link):
            Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept)   3.8668     0.9994   3.869 0.000109 ***
indcost       2.1330     0.5836   3.655 0.000257 ***
sizelog      -0.9053     0.1120  -8.082 6.38e-16 ***

Sigma coefficients (dispersion model with log link):
        Estimate Std. Error z value Pr(>|z|)
(sigma)   0.1333     0.5331    0.25    0.803

Lambda coefficient:
         Estimate Std. Error
(lambda)    1.788       1.01
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Family: PL - SLASH ( 2.29 ) (Power logit slash)
Estimation method: pML (penalized maximum likelihood)
Log-likelihood:   122 on 5 Df
Pseudo R-squared: 0.4162
Upsilon statistic: 0.06448
AIC:  -234
Number of iterations in BFGS optimization: 15
```

Normal probability plots of the residuals with simulated envelopes as well as influence plots may be obtained via `envelope()` and `influence()` functions, respectively. For instance, the plots presented in Figure 5 are obtained as follows:

```
R> envelope(Firm_slash.CD, type = "quantile")
R> envelope(Firm_slash.CD, type = "deviance")
R> envelope(Firm_slash.CD, type = "standardized")
R> influence(Firm_slash.CD)
```

Note that one observation is highlighted in almost all the graphics in Figure 5. It is the case #15 and corresponds to a firm with the highest `firmcost` value. Queiroz and Ferrari (2023b) conclude that this observation does not significantly influence the fitted model. In fact, the weight of this observation in the estimation process is close to zero — it may be verified by plotting the weights against the residuals. This plot is presented in Figure 6 and is obtained via:

```
R> plot(Firm_slash.CD, which = 7)
```

The **PLreg** package allows different link functions for the median submodel. In order to illustrate it, we fit the model with the `probit` and `cloglog` link functions; for simplicity, we set `zeta = 2.29`. We compare the fits through the values of the pseudo R^2 and the Y_ζ measure as follows:

```
R> measures <- sapply(c("logit", "probit", "cloglog"),
+                     function(x){
+                       fit <- update(Firm_slash.CD, link = x)
+                       round(c(fit$pseudo.r.squared, fit$Upsilon.zeta),3)
+                     })
R> rownames(measures) <- c("pseudo R-squared", "Upsilon_zeta")
R> measures

                logit probit cloglog
pseudo R-squared 0.416  0.379   0.482
Upsilon_zeta     0.064  0.071   0.069
```

No model simultaneously has the highest pseudo R^2 and the smallest Y_ζ. Note that the values for the fit with the `logit` and `cloglog` link functions are close. The `probit` link function leads to the smallest pseudo R^2 and the highest Y_ζ; hence it is not recommended.

4.3 `PeruVotes` data: GJS regression models

Lemonte and Bazán (2016) use the GJS Student-t regression model to model the proportion of blank votes (`votes`) in the 2006 Peruvian general election of an electoral district as a function of the Human Development Index (`HDI`). The `PeruVotes` dataset contains information on 194 electoral districts. Recall that the PL regression models with $\lambda = 1$ reduce to the GJS regression models. The extra parameter ζ of the GJS Student-t regression model may be selected using the `extra.parameter()` function as in

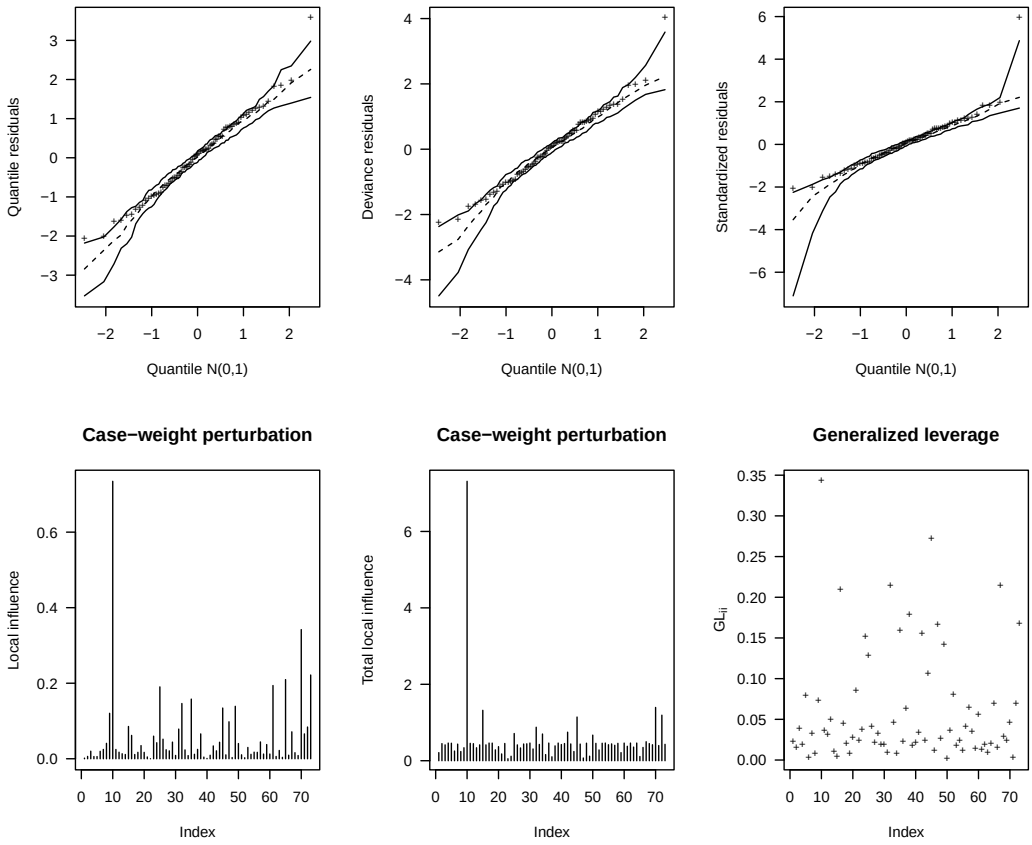

Figure 5: Normal probability plot of the quantile, deviance and standardized residuals with simulated envelope and influence plots for the `Firm_slash.CD` fit – `Firm` data.

the previous examples. However, to replicate the analysis in Lemonte and Bazán (2016) we fit the PL Student-t regression model with $\zeta = 4$ (and $\lambda = 1$). Using the `control` argument in the `PLreg()` function, we set `lambda = 1`:

```
R> data("PeruVotes", package = "PLreg")
R> PV_GJSt <- PLreg(votes ~ HDI | HDI, data = PeruVotes, family = "TF",
+                   zeta = 4, control = PLreg.control(lambda = 1))
R> summary(PV_GJSt)

Call:
PLreg(formula = votes ~ HDI | HDI, data = PeruVotes, family = "TF",
    zeta = 4, control = PLreg.control(lambda = 1))

Standardized residuals:
    Min      1Q  Median      3Q     Max
-5.1247 -0.5838  0.0024  0.5821  4.1666

Coefficients (median model with logit link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   2.3054     0.2086   11.05   <2e-16 ***
HDI          -6.8075     0.3780  -18.01   <2e-16 ***

Sigma coefficients (dispersion model with log link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.7560     0.7008  -3.933  8.4e-05 ***
HDI           2.1667     1.2422   1.744   0.0811 .

Fixed skewness parameter (lambda = 1).
---
```

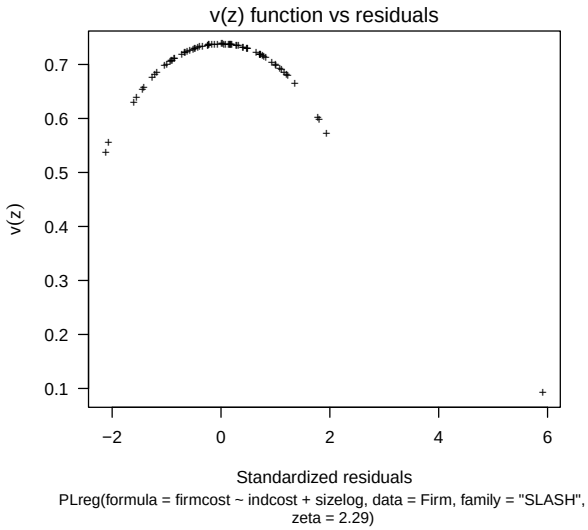

Figure 6: Plot of the $v(z)$ function against the standardized residual for the `Firm_slash.CD` fit – `Firm` data.

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Family: PL - TF ( 4 ) (Power logit Student-t)
Estimation method: ML (maximum likelihood)
Log-likelihood: 352.8 on 4 Df
Pseudo R-squared: 0.5822
Upsilon statistic: 0.06836
AIC: -697.5
Number of iterations in BFGS optimization: 9
```

As mentioned before, for fixed λ, the estimation is based on the usual maximum likelihood estimator. The output of the `summary()` function may be used to replicate part of Table 2 of Lemonte and Bazán (2016). The GJS distribution in their paper has a median-precision parameterization, while we use a median-dispersion parameterization. As we are using the logarithmic link function for the dispersion parameter, the estimates for the precision submodel are the negative of those obtained here for the dispersion submodel.

As the GJS regression models are a particular case of the PL regression models when $\lambda = 1$, an inherent question is whether the dataset supports the assumption that $\lambda = 1$. A confidence interval for λ may be constructed using the profile penalized likelihood ratio statistic defined by $W_p^*(\lambda) = 2\{\ell_p^*(\widetilde{\lambda}) - \ell_p^*(\lambda)\}$, that is asymptotically distributed as χ_1^2 (Queiroz and Ferrari, 2023b). The `CI.lambda()` function in the **PLreg** package provides a plot of $W_p^*(\lambda)$ against λ and shows the observed confidence interval for λ. As an illustration, we fit the PL Student-t regression model with $\zeta = 4$ and use the `CI.lambda()` function to obtain a 90% confidence interval for λ:

```
R> PV_PLt <- PLreg(votes ~ HDI | HDI, data = PeruVotes, family = "TF",
+                  zeta = 4)
R> coefficients(PV_PLt, conf.coef = 0.9)

       (Intercept)                 HDI (sigma)_(Intercept)
         2.3050765          -6.8065881          -2.7445500
       (sigma)_HDI            (lambda)
         2.0459250           0.9102018

R> CI.lambda(PV_PLt)

The confidence interval for lambda is: (0, 5.4).
```

The `CI.lambda()` function provides the plot presented in Figure 7; the horizontal dashed line indicates the 90% confidence interval for λ. Note that the estimated λ is close to one, and the confidence interval contains $\lambda = 1$. A diagnostic analysis not shown here indicates that the GJS Student-t regression model with $\zeta = 4$ suitably fits the data.

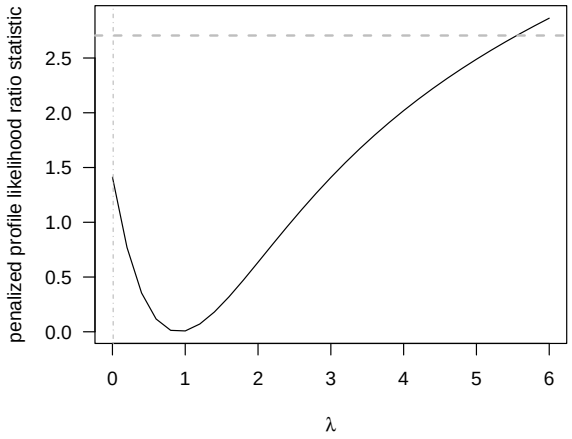

Figure 7: Plot of the profile penalized likelihood ratio statistics for λ based on the `PV_PLt` fit – `PeruVotes` data.

4.4 `bodyfat_Aeolus` data: log-log regression models

We now turn to the `bodyfat_Aeolus` data, introduced in Section 2.4.1. The interest lies in modelling the proportion of body fat of little brown bats (`percentfat`) as a function of the year (`year`, 1 for 2016 and 0 for 2009), sex of the sampled bat (`sex`, 1 for male and 0 for female) and the hibernation time (`days`), defined as the number of days since the fall equinox. First, we fit the PL normal regression model and print the estimated skewness parameter:

```
R> bodyf_PL <- PLreg(percentfat ~ days + sex + year | days + sex + year,
+                    data = bodyfat_Aeolus, family = "NO")
R> bodyf_PL$coefficients$skewness

    (lambda)
0.0007122085
```

Note that the estimate of λ is close to zero. It may indicate the limiting model when $\lambda \to 0^+$ may be reasonable. The log-log normal regression model may be fitted by setting `lambda = 0` in the `control` argument of the `PLreg()` function as follows:

```
R> bodyf_loglog <- PLreg(percentfat ~ days + sex + year | days + sex + year,
+                        data = bodyfat_Aeolus, family = "NO",
+                        control = PLreg.control(lambda = 0))
R> summary(bodyf_loglog)

Call:
PLreg(formula = percentfat ~ days + sex + year | days + sex +
    year, data = bodyfat_Aeolus, family = "NO",
    control = PLreg.control(lambda = 0))

Standardized residuals:
    Min      1Q  Median      3Q     Max
-2.7679 -0.6402  0.0664  0.6834  2.3130

Coefficients (median model with logit link):
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.1532851  0.0665887 -17.320   <2e-16 ***
days        -0.0094255  0.0005409 -17.427   <2e-16 ***
sexM        -0.0324633  0.0531725  -0.611    0.542
year2016     0.5039790  0.0581870   8.661   <2e-16 ***

Sigma coefficients (dispersion model with log link):
             Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept) -1.9668123  0.1663712 -11.822   <2e-16 ***
days         0.0007478  0.0012223   0.612   0.5407
sexM        -0.2873759  0.1145487  -2.509   0.0121 *
year2016     0.1088719  0.1314918   0.828   0.4077

Fixed skewness parameter (limiting case lambda -> 0).
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Family: log-log - NO (log-log normal)
Estimation method: ML (maximum likelihood)
Log-likelihood: 323.4 on 8 Df
Pseudo R-squared: 0.6755
Upsilon statistic: 0.05339
AIC: -630.8
Number of iterations in BFGS optimization: 28
```

Since the log-log normal regression model is more parsimonious than the PL normal regression model, it should be used. One can also consider other models in the log-log class specifying a different family in the `family` argument.

4.5 Simulated data: Inflated PL regression models

The **PLreg** package requires that the response variable values are all in the open interval $(0,1)$. It does not allow values at the boundaries, i.e., equal to zero or one. The zero-or-one inflated PL regression models may be employed when the response variable contains values at one of the boundaries. These models may be fitted using the **PLreg** package in conjunction with the `glm()` function.

We say that Y has an inflated PL distribution with parameters $\alpha \in (0,1)$, $\mu \in (0,1)$, $\sigma > 0$, and $\lambda > 0$ if $\mathbb{P}(Y = c) = \alpha$, with $c = 0$ or $c = 1$, and, with probability $1 - \alpha$, $Y \sim \text{PL}(\mu, \sigma, \lambda; r)$. In other words, an inflated PL distribution is a mixture of a PL distribution and a degenerate variable in a known value c ($c = 0$ or $c = 1$). If $c = 0$, we have the zero-inflated PL distribution and if $c = 1$, the one-inflated PL distribution. The parameters μ, σ, and λ represent the median, dispersion and skewness of the conditional distribution of Y given that $Y \in (0,1)$ and α is the mixture parameter. When $\lambda = 1$ the inflated PL distributions reduce to the inflated GJS distributions (Queiroz and Lemonte, 2021). If $\lambda \to 0^+$, we have the inflated log-log distributions as a limiting case.

In the inflated PL regression models, μ and σ are linked to the covariates through linear predictors with unknown coefficients as in Equation 1. Likewise, the mixture parameter submodel is $d_0(\alpha_i) = \boldsymbol{z}_i^\top \boldsymbol{\kappa} = \eta_{0i}$.

One may use the maximum likelihood approach to estimate the parameters of the model, denoted here by $\boldsymbol{\theta} = (\boldsymbol{\kappa}^\top, \boldsymbol{\beta}^\top, \boldsymbol{\tau}^\top, \lambda)^\top$. The likelihood function of $\boldsymbol{\theta}$ factorizes in two terms, one that depends only on $\boldsymbol{\kappa}$ (discrete part) and the other that depends on the remaining parameters (continuous part). Thus, the inference of the discrete part and the continuous part is performed separately.

We present a brief example with simulated data. We generate 300 observations from the zero-inflated PL normal regression model with a constant dispersion and logit link for the median and mixture parameter submodels:

```
R> n <- 300
R> kappa <- c(-2, 0.5)
R> beta <- c(-1.0, -2.0)
R> sigma <- 0.5
R> lambda <- 2
R> set.seed(25012023)
R> x1 <- runif(n)
R> Z <- X <- matrix(c(rep(1,n), x1), ncol = 2, byrow = FALSE)
R> alpha <- exp(Z%*%kappa)/(1 + exp(Z%*%kappa))
R> mu <- exp(X%*%beta)/(1 + exp(X%*%beta))
R> prob <- runif(n)
R> y <- ifelse((prob <= alpha), 0, rPL(n, mu, sigma, lambda, family = "NO"))
```

The histogram and boxplot of the response variable y are presented in Figure 8. We consider fitting the zero-inflated PL normal regression model in which the parameters α and μ are modeled as a function of `x1` through the logit link. To estimate the parameters associated with the discrete part, we fit a binomial regression model in which the response variable is equal to one if $y = 0$ and is equal to zero otherwise. The success probability for the i-th observation is α_i. This model is fitted using the `glm()`

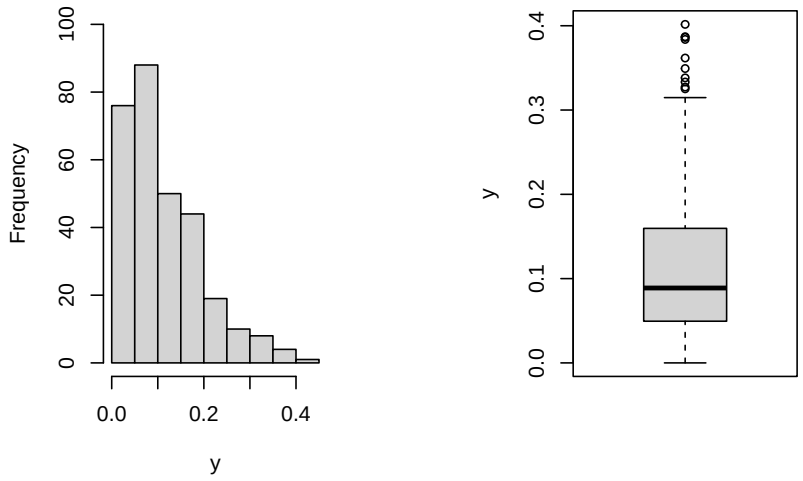

Figure 8: Histogram (left side) and boxplot (right side) of the response variable y – simulated data.

function (Chambers and Hastie, 1992) of the **stats** package. The continuous part is modeled using the **PLreg** package. The following code shows how the model is fitted:

```
R> Ind <- ifelse(y == 0, 1, 0)
R> fit.glm <- glm(Ind ~ x1, family = binomial())
R> fit.PL <- PLreg(y[Ind == 0] ~ x1[Ind == 0], family = "NO", type = "ML")
```

For estimating the parameters of the discrete part, we consider all the observations. In contrast, to estimate the parameters of the continuous part (PL model), we only consider the observations in $(0, 1)$. The estimated coefficients are obtained as follows:

```
R> coefficients(fit.glm)

(Intercept)          x1
 -2.1812682   0.9445324

R> coefficients(fit.PL)

    (Intercept)    x1[Ind == 0] (sigma)_(sigma)        (lambda)
     -0.9608803      -2.1254449       0.3261416       5.3304235
```

Standard errors and further information may be obtained through the `summary()` function. As expected, the estimates of the parameters are close to those used to generate the data. Diagnostic plots for the discrete and continuous parts may be obtained separately by using the `plot()` method for the `fit.glm` and `fit.PL` fits. The overall adequacy of the fitted model may be investigated using the randomized quantile residual (Dunn and Smyth, 1996). For the inflated PL regression models, the randomized quantile residuals are defined as:

$$r_i = \begin{cases} \Phi^{-1}(u_i), & y_i = c, \\ \Phi^{-1}\left(\widetilde{\alpha}_i \mathbb{I}_{[c,\infty)}(y) + (1-\widetilde{\alpha}_i) F_Y(y_i; \widetilde{\mu}_i, \widetilde{\sigma}_i, \widetilde{\lambda})\right), & y_i \in (0,1), \end{cases}$$

for $i = 1, \ldots, n$, where $c = 0$ or $c = 1$ depending on the case. Also, u_i is a random draw from the uniform distribution on the interval $(0, \widetilde{\alpha}_i)$ if $c = 0$, and $(1 - \widetilde{\alpha}_i, 1)$ if $c = 1$. As the **PLreg** package provides the `pPL()` function to obtain the cdf of the PL distributions, the randomized quantile residuals can be easily computed. For the data under investigation, the code below provides the computation of the residuals and the plots presented in Figure 9.

```
R> alpha  <- fit.glm$fitted.values
R> mu     <- fit.PL$link$median$linkinv(X%*%fit.PL$coefficients$median)
R> sigma  <- fit.PL$link$dispersion$linkinv(fit.PL$coefficients$dispersion)
R> lambda <- fit.PL$coefficients$skewness
R> cdf <- alpha*as.numeric(y >= 0) + (1 - alpha)*pPL(y, mu, sigma, lambda,
```

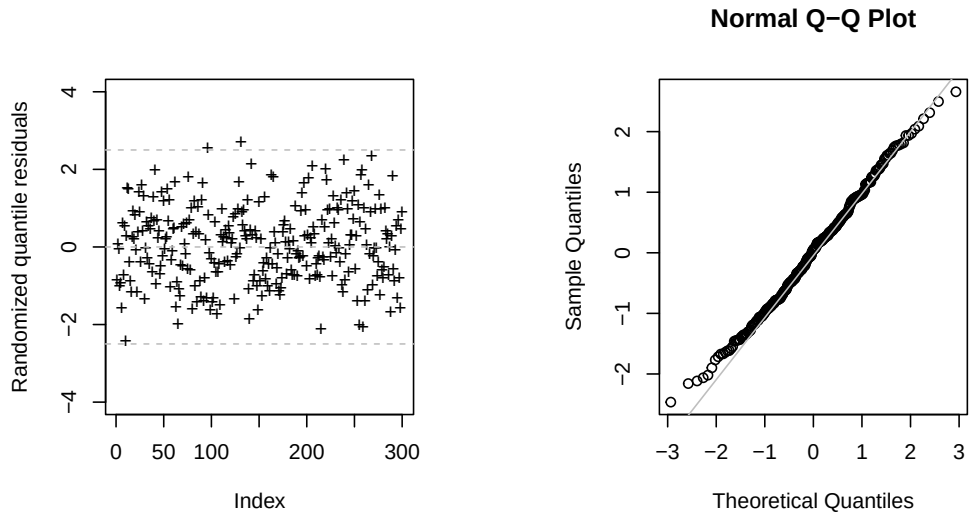

Figure 9: Scatter plot (left side) and quantile-quantile plot (right side) of the randomized quantile residual – simulated data.

```
+        family = "NO")
R> res <- ifelse(y == 0, qnorm(runif(length(y), 0, alpha)), qnorm(cdf))
R> plot(res, ylab = "Randomized quantile residuals", pch = "+", ylim = c(-4, 4))
R> abline(h = 2.5, col = "gray", lty = 2)
R> abline(h = -2.5, col = "gray", lty = 2)
R> abline(h = 0, col = "gray", lty = 2)
R> qqnorm(res)
R> qqline(res, col = "gray")
```

We may also fit inflated GJS and inflated log-log regression models specifying `lambda = 1` and `lambda = 0` in the `control` argument of the `PLreg()` function, respectively.

5 Concluding remarks

This paper presents the R implementation of the PL regression models available in the **PLreg** package. The models are suitable for modeling continuous data observed in the open interval (0,1). The package provides tools for likelihood-based inference and diagnostic analysis. Currently, the package includes seven distributions in the PL class, two types of estimators, profile likelihood-based confidence intervals for the skewness parameter, and procedures for selecting the extra parameter, if any. Different residuals and influence methods for performing diagnostic analysis are implemented. The applications in the previous sections illustrate the ability of the package to fit different PL regression models, including the GJS and log-log models.

The response variable for using the **PLreg** package must lie in the open interval (0,1), as it is an inherent assumption of the PL regression models. A possible approach when the data contain observations in one of the boundaries is to employ the inflated PL regression models, that assume that the response variable has a mixture of a PL distribution and a degenerate distribution at zero or one. A relevant contribution of this paper is to show how the **PLreg** package can be used to fit and perform diagnostic analysis for inflated PL regression models as well as inflated GJS and log-log regression models.

Acknowledgments

We thank the associate editor and the reviewer for their constructive comments on an earlier version of this article. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001 and by the Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brazil (CNPq). The authors gratefully acknowledge funding provided by CNPq (Grants No. 150976-2022-4 and No. 305963-2018-0).

References

O. E. Barndorff-Nielsen and B. Jørgensen. Some parametric models on the simplex. *Journal of Multivariate Analysis*, 39(1):106–116, 1991. doi: 10.1016/0047-259X(91)90008-P. [p235]

C. L. Bayes, J. L. Bazán, and C. García. A new robust regression model for proportions. *Bayesian Analysis*, 7(4):841 – 866, 2012. doi: 10.1214/12-BA728. [p235]

J. M. Chambers and T. J. Hastie, editors. *Statistical Models in S*. Chapman & Hall, London, 1992. [p250]

T. L. Cheng, A. Gerson, M. S. Moore, J. D. Reichard, J. DeSimone, C. K. R. Willis, W. F. Frick, and A. M. Kilpatrick. Higher fat stores contribute to persistence of little brown bat populations with white-nose syndrome. *Journal of Animal Ecology*, 88(4):591–600, 2019. doi: 10.1111/1365-2656.12954. [p239]

R. F. da Paz, N. Balakrishnan, and J. L. Bazán. L-logistic regression models: Prior sensitivity analysis, robustness to outliers and applications. *Brazilian Journal of Probability and Statistics*, 33(3):455 – 479, 2019. doi: 10.1214/18-BJPS397. [p236]

P. K. Dunn and G. K. Smyth. Randomized quantile residuals. *Journal of Computational and Graphical Statistics*, 5(3):236–244, 1996. doi: 10.1080/10618600.1996.10474708. [p250]

S. L. P. Ferrari and F. Cribari-Neto. Beta regression for modelling rates and proportions. *Journal of Applied Statistics*, 31(7):799–815, 2004. doi: 1010.1080/0266476042000214501. [p235]

E. Gómez-Déniz, M. A. Sordo, and E. Calderín-Ojeda. The log–lindley distribution as an alternative to the beta regression model with applications in insurance. *Insurance: Mathematics and Economics*, 54: 49–57, 2014. doi: 10.1016/j.insmatheco.2013.10.017. [p235]

N. L. Johnson. Systems of frequency curves generated by methods of translation. *Biometrika*, 36(1/2): 149–176, 1949. doi: 10.2307/2332539. [p236]

M. C. Korkmaz. A new heavy-tailed distribution defined on the bounded interval: The logit slash distribution and its application. *Journal of Applied Statistics*, 47(12):2097–2119, 2020. doi: 10.1080/02664763.2019.1704701. [p236]

A. J. Lemonte and J. L. Bazán. New class of johnson distributions and its associated regression model for rates and proportions. *Biometrical Journal*, 58(4):727–746, 2016. doi: 10.1002/bimj.201500030. [p235, 236, 245, 246, 247]

F. F. Queiroz and S. L. P. Ferrari. *PLreg: Power Logit Regression for Modeling Bounded Data*, 2023a. URL https://CRAN.R-project.org/package=PLreg. R package version 0.4.1. [p239]

F. F. Queiroz and S. L. P. Ferrari. Power logit regression for modeling bounded data. *Statistical Modelling*, 2023b. doi: 10.1177/1471082X221140157. [p235, 236, 237, 239, 240, 243, 244, 245, 247]

F. F. Queiroz and A. J. Lemonte. A broad class of zero-or-one inflated regression models for rates and proportions. *Canadian Journal of Statistics*, 49(2):566–590, 2021. doi: 10.1002/cjs.11576. [p249]

J. T. Schmit and K. Roth. Cost effectiveness of risk management practices. *The Journal of Risk and Insurance*, 57(3):455–470, 1990. [p243]

Y. Shou and M. Smithson. *cdfquantreg: Quantile Regression for Random Variables on the Unit Interval*, 2022. URL https://CRAN.R-project.org/package=cdfquantreg. R package version 1.3.1-1. [p235]

M. Smithson and Y. Shou. Cdf-quantile distributions for modelling random variables on the unit interval. *British Journal of Mathematical and Statistical Psychology*, 70(3):412–438, 2017. doi: 10.1111/bmsp.12091. [p235]

D. M. Stasinopoulos and R. A. Rigby. Generalized additive models for location scale and shape (GAMLSS) in r. *Journal of Statistical Software*, 23(7):1–46, 2007. doi: 10.18637/jss.v023.i07. [p235]

A. Zeileis and Y. Croissant. Extended model formulas in r: Multiple parts and multiple responses. *Journal of Statistical Software*, 34(1):1–13, 2010. doi: 10.18637/jss.v034.i01. [p239]

A. Zeileis, F. Cribari-Neto, B. Gruen, and I. Kosmidis. *betareg: Beta Regression*, 2021. URL https://CRAN.R-project.org/package=betareg. R package version 3.1-4. [p235]

P. Zhang and Z. Qiu. Regression analysis of proportional data using simplex distribution. *Science China Mathematics (Chinese Version)*, 44(1):89–104, 2014. doi: 10.1360/012013-200. [p235]

P. Zhang, Z. Qiu, and C. Shi. *simplexreg: Regression Analysis of Proportional Data Using Simplex Distribution*, 2016. URL https://CRAN.R-project.org/package=simplexreg. R package version 1.3. [p235]

Francisco F. Queiroz
Department of Statistics, University of São Paulo
Rua do Matão, 1010
05508-090, São Paulo, Brazil
*E-mail: email:*felipeq@ime.usp.br

Silvia L.P. Ferrari
Department of Statistics, University of São Paulo
Rua do Matão, 1010
05508-090, São Paulo, Brazil
*E-mail: email:*silviaferrari@usp.br