

Package ‘lmls’

November 20, 2024

Title Gaussian Location-Scale Regression

Version 0.1.1

Description The Gaussian location-scale regression model is a multi-predictor model with explanatory variables for the mean (= location) and the standard deviation (= scale) of a response variable. This package implements maximum likelihood and Markov chain Monte Carlo (MCMC) inference (using algorithms from Girolami and Calderhead (2011) <[doi:10.1111/j.1467-9868.2010.00765.x](https://doi.org/10.1111/j.1467-9868.2010.00765.x)> and Nesterov (2009) <[doi:10.1007/s10107-007-0149-x](https://doi.org/10.1007/s10107-007-0149-x)>), a parametric bootstrap algorithm, and diagnostic plots for the model class.

License MIT + file LICENSE

URL <https://hriebl.github.io/lmls/>, <https://github.com/hriebl/lmls>

BugReports <https://github.com/hriebl/lmls/issues>

Depends R (>= 3.5.0)

Imports generics (>= 0.1.0)

Suggests bookdown, coda, covr, ggplot2, knitr, mgcv, mvtnorm, numDeriv, patchwork, rmarkdown, testthat (>= 3.0.0)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Hannes Riebl [aut, cre]

Maintainer Hannes Riebl <hriebl@posteo.de>

Repository CRAN

Date/Publication 2024-11-20 22:20:02 UTC

Contents

abdom	2
boot	3
lmls	4
lmls-methods	5
mcmc	6
summary.lmls	7
Index	9

abdom	<i>Abdominal circumference data</i>
-------	-------------------------------------

Description

The abdom data frame has 610 rows and 2 columns. The data are measurements of abdominal circumference (response variable) taken from fetuses during ultrasound scans at Kings College Hospital, London, at gestational ages (explanatory variable) ranging between 12 and 42 weeks.

Usage

```
data(abdom)
```

Format

This data frame contains the following columns:

- y Abdominal circumference: a numeric vector.
- x Gestational age: a numeric vector.

Details

The data were used to derive reference intervals by Chitty *et al.* (1994) and also for comparing different reference centile methods by Wright and Royston (1997), who also commented that the distribution of z-scores obtained from the different fitted models "has somewhat longer tails than the normal distribution".

Source

Dr. Eileen M. Wright, Department of Medical Statistics and Evaluation, Royal Postgraduate Medical School, Du Cane Road, London, W12 0NN.

The abdom dataset was copied into the lmls package from the gamlss.data package. gamlss.data is licensed under the GPL 2 and 3.

References

Chitty, L.S., Altman, D.G., Henderson, A. and Campbell, S. (1994). Charts of fetal size: 3, abdominal measurement. *Br. J. Obstet. Gynaec.*, **101**: 125–131.

Wright, E.M. and Royston, P. (1997). A comparison of statistical methods for age-related reference intervals. *J. R. Statist. Soc. A.*, **160**: 47–69.

Examples

```
data(abdom)
attach(abdom)
plot(x, y)
detach(abdom)
```

boot

Parametric bootstrap for LMLS

Description

A simple parametric bootstrap algorithm for location-scale regression models from the [lmls\(\)](#) function.

Usage

```
boot(m, num_samples = 1000, seed = NULL)
```

Arguments

<code>m</code>	A location-scale regression model from the lmls() function.
<code>num_samples</code>	The number of bootstrap samples to draw.
<code>seed</code>	Either NULL or an integer that will be used in a call to set.seed() before simulating the response vectors.

Value

An `lmls` S3 object, see [lmls\(\)](#). The entry `boot` with the matrices of bootstrap samples is added to the object as a list with the names `location` and `scale`.

Examples

```
library(lmls)
m <- lmls(y ~ poly(x, 2), ~ x, data = abdom, light = FALSE)
m <- boot(m)
summary(m, type = "boot")
hist(m$boot$scale[, 2])
```

lmls

Gaussian location-scale regression

Description

The location-scale regression model assumes a normally distributed response variable with one linear predictor for the mean (= the location) and one for the standard deviation (= the scale). The standard deviation is mapped to the linear predictor through a log link.

This function sets up the model object and estimates it with maximum likelihood.

Usage

```
lmls(
  location,
  scale = ~1,
  data = environment(location),
  light = TRUE,
  maxit = 100,
  reltol = sqrt(.Machine$double.eps)
)
```

Arguments

location	A two-sided formula with the response variable on the LHS and the predictor for the mean on the RHS.
scale	A one-sided formula with the predictor for the standard deviation on the RHS.
data	A data frame (or list or environment) in which to evaluate the location and scale formulas.
light	If TRUE, the design matrices are removed from the estimated model to save some memory.
maxit	The maximum number of iterations of the Fisher scoring algorithm.
reltol	The relative convergence tolerance of the Fisher scoring algorithm.

Value

A fitted linear model for location and scale as an `lmls` S3 object. The object has at least the following entries:

- `y`: the response vector
- `nobs`: the number of observations
- `df`: the degrees of freedom
- `df.residual`: the residual degrees of freedom
- `coefficients`: the regression coefficients as a list with the names `location` and `scale`
- `fitted.values`: the fitted values as a list with the names `location` and `scale`

- residuals: the response residuals
- coefficients: the variance-covariance matrices of the regression coefficients as a list with the names location and scale
- iterations: the number of iterations the Fisher scoring algorithm took to converge

Examples

```
library(lmls)
m <- lmls(y ~ poly(x, 2), ~ x, data = abdom)
summary(m)
plot(m)
qqnorm(m)
```

lmls-methods

Methods for LMLS

Description

A couple of methods for location-scale regression models from the `lmls()` function are provided.

Usage

```
## S3 method for class 'lmls'
coef(object, predictor = c("location", "scale"), ...)

## S3 method for class 'lmls'
fitted(object, predictor = c("location", "scale"), ...)

## S3 method for class 'lmls'
predict(
  object,
  newdata = NULL,
  predictor = c("location", "scale"),
  type = c("link", "response"),
  ...
)

## S3 method for class 'lmls'
residuals(object, type = c("deviance", "pearson", "response"), ...)

## S3 method for class 'lmls'
vcov(object, predictor = c("location", "scale"), ...)
```

Arguments

object	A location-scale regression model from the <code>lmls()</code> function.
predictor	The predictor to work on. Either "location" or "scale" or both. If both, a list with the names "location" and "scale" is returned.
...	Currently ignored.
newdata	A data frame (or list or environment) with the covariate values at which the predictions are computed. If NULL, the predictions at the original data are returned.
type	Used by <code>predict()</code> and <code>residuals()</code> : <ul style="list-style-type: none"> • For <code>predict()</code>, "link" or "response". If "link" (default), μ and $\log(\sigma)$ are returned. If "response", μ and σ are returned. • For <code>residuals()</code>, "deviance", "pearson" or "response". If "deviance" (default) or "pearson", $(y - \mu) / \sigma$ is returned. If "response", $y - \mu$ is returned.

Value

A numeric vector for `residuals()`. For the other methods, a numeric vector if the argument predictor is either "location" or "scale", or a list with the names location and scale if it is both.

mcmc

MCMC inference for LMLS

Description

A Markov chain Monte Carlo (MCMC) sampler for location-scale regression models from the `lmls()` function. The sampler uses Gibbs updates for the location coefficients and the Riemann manifold Metropolis-adjusted Langevin algorithm (MMALA) from Girolami and Calderhead (2011) with the Fisher-Rao metric tensor for the scale coefficients. The priors for the regression coefficients are assumed to be flat.

To find the optimal step size for the MMALA updates, the dual averaging algorithm from Nesterov (2009) is used during a warm-up phase.

Usage

```
mcmc(m, num_samples = 1000, num_warmup = 1000, target_accept = 0.8)
```

Arguments

m	A location-scale regression model from the <code>lmls()</code> function.
num_samples	The number of MCMC samples after the warm-up. Defaults to 1000.
num_warmup	The number of MCMC samples for the warm-up. Defaults to 1000.
target_accept	The target acceptance rate for the dual averaging algorithm used for the warm-up. Defaults to 0.8.

Value

An lmls S3 object, see `lmls()`. The entry `mcmc` with the matrices of MCMC samples is added to the object as a list with the names `location` and `scale`.

References

Girolami, M. and Calderhead, B. (2011), Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73: 123-214. doi:10.1111/j.14679868.2010.00765.x

Nesterov, Y. (2009), Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120: 221–259. doi:10.1007/s101070070149x

Examples

```
library(lmls)
m <- lmls(y ~ poly(x, 2), ~ x, data = abdom, light = FALSE)
m <- mcmc(m)
summary(m, type = "mcmc")
plot(m$mcmc$scale[, 2], type = "l")
```

summary.lmls

*Summary for LMLS***Description**

Prints a summary for location-scale regression models from the `lmls()` function.

Usage

```
## S3 method for class 'lmls'
summary(
  object,
  type = c("ml", "boot", "mcmc"),
  digits = max(3, getOption("digits") - 3),
  ...
)
```

Arguments

<code>object</code>	A location-scale regression model from the <code>lmls()</code> function.
<code>type</code>	Either "ml" or "boot" or "mcmc": <ul style="list-style-type: none"> • If "ml", the maximum likelihood estimates and the asymptotic standard errors are shown. • If "boot", the bootstrap estimates and confidence intervals are shown. • If "mcmc", the Markov chain Monte Carlo (MCMC) estimates and credible intervals are shown.
<code>digits</code>	The number of digits to print.
<code>...</code>	Passed on to <code>printCoefmat()</code> .

Value

The (unmodified and invisible) `lmls` S3 object, see [lmls\(\)](#).

Index

`abdom`, [2](#)

`boot`, [3](#)

`coef.lmls (lmls-methods)`, [5](#)

`fitted.lmls (lmls-methods)`, [5](#)

`lmls`, [4](#)

`lmls()`, [3](#), [5–8](#)

`lmls-methods`, [5](#)

`mcmc`, [6](#)

`predict.lmls (lmls-methods)`, [5](#)

`printCoefmat()`, [7](#)

`residuals.lmls (lmls-methods)`, [5](#)

`set.seed()`, [3](#)

`summary.lmls`, [7](#)

`vcov.lmls (lmls-methods)`, [5](#)