

# Package ‘logNormReg’

October 16, 2019

**Type** Package

**Title** log Normal Linear Regression

**Version** 0.3-0

**Date** 2019-10-15

**Maintainer** Vito M. R. Muggeo <vito.muggeo@unipa.it>

**Description** Functions to fits simple linear regression models with log normal errors and identity link (taking the responses on the original scale). See Muggeo (2018) <doi:10.13140/RG.2.2.18118.16965>.

**License** GPL

**NeedsCompilation** no

**Author** Vito M. R. Muggeo [aut, cre] (<<https://orcid.org/0000-0002-3386-4054>>)

**Repository** CRAN

**Date/Publication** 2019-10-16 18:00:02 UTC

## R topics documented:

logNormReg-package . . . . .	2
confint.lognlm . . . . .	3
logLik.lognlm . . . . .	4
lognlm . . . . .	6
lognlm.fit . . . . .	8
print.lognlm . . . . .	9
summary.lognlm . . . . .	10
vcov.lognlm . . . . .	11

<b>Index</b>	<b>13</b>
--------------	-----------

---

logNormReg-package      *log Normal Linear Regression*

---

### Description

Functions to perform and to display results from simple multiple linear regression with log Normal errors and identity link. Standard errors of estimates are based on the hessian or alternatively on the sandwich formula.

### Details

Package: logNormReg  
Type: Package  
Version: 0.2-0  
Date: 2019-10-15  
License: GPL

### Acknowledgement

This package was inspired by a fruitful discussion with Andrew Beet (Marine Policy Center, Woods Hole Oceanographic Institution, U.S.).

### Author(s)

Vito M.R. Muggeo <vito.muggeo@unipa.it>

### References

Gustavsson, S., Fagerberg, B., Sallsten, G., Andersson, E. M. (2014). Regression Models for Log-Normal Data: Comparing Different Methods for Quantifying the Association between Abdominal Adiposity and Biomarkers of Inflammation and Insulin Resistance . *International Journal of Environmental Research and Public Health*, **11**, 3521–3539.

Muggeo, V.M.R. (2018) A note on regression with log Normal errors: linear and piecewise linear modelling in R, doi: 10.13140/RG.2.2.18118.16965.

### See Also

See [lognlm](#) for the main function with a toy example.

See also package [gamLss](#) for more general regression models including log Normal errors.

---

confint.lognlm                      *Confidence intervals for the parameters in log normal regression*

---

### Description

Computes confidence intervals (based on the Gradient, Wald or Likelihood Ratio statistic) for the linear parameters in a fitted 'lognreg' model.

### Usage

```
## S3 method for class 'lognlm'
confint(object, parm, level=0.95, type = c("gradient", "wald", "lrt"), ...)
```

### Arguments

object	a fitted lognreg object.
parm	the parameter of interest. Numeric (covariate number) or character (covariate name).
level	the required confidence level (default to 0.95).
type	Which statistics should be used? Currently "wald", "gradient", or "lrt". Names can be abbreviated. If object has been obtained with lik=FALSE, only type="wald" is permitted.
...	Other optional arguments to be passed on the internal functions. Ignored if type="wald", otherwise these can be: <ul style="list-style-type: none"> <li>- lim to specify the range of the evaluation points (default to (-3,3) resulting in the interval <math>\hat{\beta} \pm 3 \times SE(\hat{\beta})</math>);</li> <li>- values to set explicitly the evaluation point(s);</li> <li>- return.val to return (if TRUE) the evaluation points and the corresponding statistic values (useful to plot the profiled statistic). If the supplied values includes just one scalar, return.val is set to TRUE.</li> </ul>

### Details

Confidence intervals are computed and returned. Currently only the Wald, Gradient or Likelihood ratio statistic can be used.

### Value

The end-points of confidence intervals.

### Author(s)

Vito Muggeo

## References

For a gentle and general introduction about the likelihood-based statistics (including the gradient) see

Muggeo V.M.R., Lovison G. (2014), The 'three plus one' likelihood-based test statistics: unified geometrical and graphical interpretations. *The American Statistician*, **68**, 302-306.

## See Also

[lognlm](#)

## Examples

```
n=50
s=.4
set.seed(1515)      #just to get reproducible results..

#covariates
x<-seq(.1,10,l=n)
z<-rnorm(n)

#response
mu<- 10+.5*x- z #linear regression function
y<-rlnorm(n, log(mu)-s^2/2, s) #data..

o<- lognlm(y~x+z) #the model

confint(o, "x", type="g")
confint(o, "z", type="w") #same than confint.default(o)
```

---

logLik.lognlm

*Log Likelihood for log Normal linear regression*

---

## Description

The function returns the log-likelihood value of the log Normal linear regression model evaluated at the estimated coefficients

## Usage

```
## S3 method for class 'lognlm'
logLik(object, full=FALSE, ...)
## S3 method for class 'lognlm'
extractAIC(fit, scale=0, k=2, ...)
```

**Arguments**

object, fit	A lognlm fit returned by lognlm()
full	If FALSE, only the <i>kernel</i> of the log likelihood is returned, otherwise the complete log likelihood (including terms depending on data only)
scale	Optional numeric specifying the scale parameter of the model. <code>scale = 0</code> indicates that it is to be estimated by maximum likelihood.
k	Optional numeric specifying the penalty of the edf in the AIC formula.
...	optional arguments (nothing in this method).

**Details**

If object has been obtained via `lognlm(...,lik=TRUE)`, `logLik.lognlm` returns the log likelihood (kernel or complete, depending on argument `full`), otherwise the sum of log residuals,  $\sum((\log(y_i) - \log(\hat{\mu}_i))^2)$ .

**Value**

The log likelihood (or the sum of log residuals squared) of the model fit object

**Author(s)**

Vito Muggeo

**See Also**

[lognlm](#)

**Examples**

```
# o is the fit object, see ?lognlm
n=50
s=.4

#covariates
x<-seq(.1,10,l=n)

#response
set.seed(1234)      #just to get reproducible results..
mu<- 10+.5*x #linear regression function
y<-rlnorm(n, log(mu)-s^2/2, s) #data..

o<- lognlm(y~x) #the model

logLik(o) #the kernel log likelihood value
logLik(o, full=TRUE)
```

lognlm

*Multiple linear regression with log Normal errors***Description**

The function fits simple multiple linear regression models with log Normals erros. Two objectives as well as two optimizing functions can be used.

**Usage**

```
lognlm(formula, data, subset, weights, na.action, y = TRUE, start, model = TRUE,
       lik = TRUE, opt = c("nlminb", "optim"), contrasts=NULL, ...)
```

**Arguments**

formula	a standard R formula with response and explanatory variables (and possible off-set) specifying the regression model being fitted.
data	an optional data frame, list or environment containing some or all the variables in the model.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
weights	an optional vector of weights to be used in the fitting process. Currently unimplemented.
na.action	an optional vector of weights to be used in the fitting process.
y	logical. If TRUE the response vector is returned as y in the fit object.
start	(optional) starting values of the parameter to be estimated. If start is missing they are computed via ordinary least squares with the intercept $\hat{\beta}_0$ replaced by $\max(\hat{\beta}_0, \text{median}\{y_i\})$ . If lik=TRUE (i.e. a log Normal model is fitted), start refers to the regression parameters <i>and</i> the error standard deviation; if lik=FALSE, start does <i>not</i> include the starting guess for the standard deviation.
model	logical. If TRUE the model frame is returned as model in the fit object.
lik	If TRUE the log Normal log likelihood is optimized, otherwise the sum of squared residuals based on the logs (see Details).
opt	the optimization function to be used. nlminb has shown to be more efficient, probably because it uses (unlike optim) also the hessian matrix (supplied in the code).
contrasts	an optional list. See the contrasts.arg of model.matrix.default.
...	optional arguments passed on to the optimizing functions (nlminb or optim), (and therefore should be <i>consistent</i> with that).

**Details**

lognlm fits simple linear regression models with log Normal errors and identity link. Actually two objectives could be used.

If `lik=TRUE` the usual log Normal likelihood is optimized, otherwise estimation is based on minimization of the following loss function

$$\sum_i (\log y_i - \log \mu_i)^2$$

where  $\mu_i = x_i^T \beta$  is the mean function equal to the linear predictor (as an identity link is exploited).

**Value**

A list with components

<code>coefficients</code>	the regression parameters estimate.
<code>loglik</code>	The objective function value, namely the log Normal log likelihood or the sum of the squared 'log residuals' (depending on <code>lik</code> option).
<code>s2</code>	the error variance estimate.
<code>fitted.values</code>	the fitted values.
<code>residuals</code>	the raw residuals, i.e. 'observed - fitted'.
<code>grad</code>	the gradient at solution.
<code>hessian</code>	the hessian matrix at solution.
<code>convergence</code>	the convergence code coming from the fitter function. 0 means successful convergence.
<code>call</code>	the matched call.
<code>y</code>	the response vector (provided that <code>y=TRUE</code> has been set).
<code>opt</code>	the employed optimizer.
<code>lik</code>	logical, indicating if the fit comes from a log Normal likelihood approach.
<code>xlevels</code>	(only where relevant) a record of the levels of the factors used in fitting.
<code>terms</code>	the terms object used.
<code>contrasts</code>	(only where relevant) the contrasts used.
<code>model</code>	if requested, i.e. <code>model=TRUE</code> has been set (the default), the model frame used.
<code>offset</code>	the (possible) offset used.

**Author(s)**

Vito M.R. Muggeo

**See Also**

See also [print.lognlm](#) and [summary.lognlm](#) to display results.

## Examples

```

n=200
s=.4
set.seed(1234)      #just to get reproducible results..

x<-seq(.1,10,l=n) #covariate
mu<- 10+2*x      #linear regression function
y<-rlnorm(n, log(mu)-s^2/2, s) #data..

o0<-lm(log(y)~x) #the usual but WRONG model
o<- lognlm(y~x) #fit the 'right' model

plot(x,y)
lines(x, mu, lwd=2)
points(x, exp(fitted(o0)), col=2, type="l", lwd=2)
points(x, fitted(o), col=3, type="l", lwd=2)
legend("topleft", legend=c("true", "lm(log(y)~x)", "lognlm(y~x)"),
      col=c(1,2,3), lwd=2)

#Sometimes people would estimate parameters by minimizing a least square objective
# (i.e. by setting 'lik=FALSE', see Details), wherein data would come from
# Y = mu * exp(eps) where eps~N(0,s)..
y1<-mu*exp(rnorm(n,0,1)) #data..
o1<-lognlm(y1~x, lik=FALSE) #set 'lik=FALSE', see Details

```

---

lognlm.fit

*The fitter function for log Normal Linear Models*


---

## Description

lognlm.fit is called by lognlm to fit log Normal linear regression models. Two optimizing functions can be used, nlminb and optim. This function is not meant to be called by the user directly.

## Usage

```
lognlm.fit(X, y, par, lik = TRUE, opt = c("nlminb", "optim"), offset=NULL, ...)
```

## Arguments

X	design matrix for standard linear terms.
y	vector of observations of length n.
par	starting values of parameters to be estimated.
lik	logical. See lik in <a href="#">lognlm</a>
opt	the optimizing algorithm. Default to nlminb.
offset	a possible offset term.
...	other arguments to be passed to the optimizer specified in opt.



**Details**

See [lognlm](#) for more details on the arguments and returned objects.

**Value**

A list of fit information

**Note**

This function should usually not be used directly by the user.

**Author(s)**

Vito M.R. Muggeo

**See Also**

[nlminb](#), [optim](#), [lognlm](#)

**Examples**

```
## See ?lognlm
```

---

`print.lognlm`      *Print method for the lognlm class*

---

**Description**

Printing the most important features of a 'lognlm' model.

**Usage**

```
## S3 method for class 'lognlm'  
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

**Arguments**

<code>x</code>	object of class segmented
<code>digits</code>	number of digits to be printed
<code>...</code>	arguments passed to other functions

**Author(s)**

Vito M.R. Muggeo

**See Also**

[summary.lognlm](#), [print.summary.lognlm](#)

---

summary.lognlm            *Summarizing model fits for log Normal regression*

---

### Description

summary method for class lognlm.

### Usage

```
## S3 method for class 'lognlm'  
summary(object, ...)  
  
## S3 method for class 'summary.lognlm'  
print(x, digits = max(3L, getOption("digits") - 3L),  
      signif.stars = getOption("show.signif.stars"), ...)
```

### Arguments

object	object of class "lognreg".
x	a summary.segmented object produced by summary.segmented().
digits	controls number of digits printed in output.
signif.stars	logical, should stars be printed on summary tables of coefficients?
...	further arguments to be passed to vcov, for instance sandwich=TRUE.

### Details

These functions compute and print some useful information relevant to "lognlm" fits, including point estimates, standard errors and p-values.

### Value

A list (similar to one returned by lognlm with additional components, such as the estimate standard errors and corresponding p-values).

### Author(s)

Vito Muggeo

### See Also

See also [lognlm](#) and [vcov.lognlm](#)

**Examples**

```
## Not run:
n=20
s=.2
set.seed(10)      #just to get reproducible results..

#covariates
x<-seq(.1,10,l=n)
z<-rnorm(n)

#response
mu<- 10+.5*x- z #linear regression function
y<-rlnorm(n, log(mu)-s^2/2, s) #data..

o<- lognlm(y~x+z) #the model
summary(o, sandw=TRUE)

## End(Not run)
```

---

vcov.lognlm	<i>Covariance matrix for lognlm fits</i>
-------------	--

---

**Description**

Computes covariance matrix of parameter estimates from a lognlm fit.

**Usage**

```
## S3 method for class 'lognlm'
vcov(object, sandw = FALSE, emp = TRUE, se = FALSE, ...)
```

**Arguments**

object	a fitted model object of class "lognlm" returned by lognlm()
sandw	logical, if TRUE the returned covariance matrix comes from the sandwich formula, otherwise it is just the (minus) hessian. Meaningless if object has been obtained with lik=FALSE.
emp	logical; if TRUE (and sandw=TRUE too), the 'meat' (i.e the information matrix) is computed empirically by the outer product of the individual score contributions.
se	logical; if TRUE the square root of the elements of the main diagonal are returned (rather than the whole matrix).
...	additional arguments.

**Details**

If object has been obtained via lognlm(..., lik=TRUE) the returned covariance matrix (or standard errors only) refers to regression coefficients *and* the response standard deviation. Otherwise (if lik=FALSE has been set), it includes entries relevant to regression coefficients.

**Value**

The variance-covariance matrix of the parameter estimates, if `se=FALSE`; otherwise the square root of the main diagonal entries.

**Author(s)**

Vito Muggeo

**See Also**

[lognlm](#)

**Examples**

```
n=50
s=.3
set.seed(1515)      #just to get reproducible results..

#covariates
x<-seq(.1,10,l=n)
z<-rnorm(n)

#response
mu<- 10+.5*x- z #linear regression function
y<-rlnorm(n, log(mu)-s^2/2, s) #data..

o<- lognlm(y~x+z) #the model
vcov(o) #the full covariance matrix
vcov(o, se=TRUE, sandw=TRUE) #st.errs only, based on the sandwich formula

o<- lognlm(y~x+z, lik=FALSE) #min. distance estimation
vcov(o, se=TRUE) #'sandw=TRUE' is not allowed if the fit has been obtained with 'lik=FALSE'
```

# Index

- \*Topic **models**
  - print.lognlm, 9
- \*Topic **nonlinear**
  - confint.lognlm, 3
- \*Topic **package**
  - logNormReg-package, 2
- \*Topic **regression**
  - confint.lognlm, 3
  - logLik.lognlm, 4
  - lognlm, 6
  - lognlm.fit, 8
  - summary.lognlm, 10
  - vcov.lognlm, 11

confint.lognlm, 3

extractAIC.lognlm(logLik.lognlm), 4

logLik.lognlm, 4

lognlm, 2, 4, 5, 6, 8–10, 12

lognlm.fit, 8

logNormReg(logNormReg-package), 2

logNormReg-package, 2

nlminb, 9

optim, 9

print.lognlm, 7, 9

print.summary.lognlm, 9

print.summary.lognlm(summary.lognlm),  
10

summary.lognlm, 7, 9, 10

vcov.lognlm, 10, 11