Package: logNormReg (via r-universe)

September 17, 2024

Type Package

Title log Normal Linear Regression

Version 0.5-0

Date 2021-11-08

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

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

Depends R (>= 3.5.0)

License GPL

NeedsCompilation no

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

Date/Publication 2021-11-08 17:00:02 UTC

Repository https://vitomuggeo.r-universe.dev

RemoteUrl https://github.com/cran/logNormReg

RemoteRef HEAD

RemoteSha a5c9af006d98a3778090f4478ba8cdc470494853

Contents

`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 sandwich formula.

Details

Package:	logNormReg
Type:	Package
Version:	0.5-0
Date:	2021-11-08
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 sattistic) for the linear parameters in a fitted 'lognreg' model.

Usage

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

Arguments

`object`	a fitted `lognlm` object.
`parm`	the parameter of interest. Numeric (covariate number) or character (covariate name). If missing `parm=2` is taken, i.e. the coefficient of the first covariate, provided the intercept is in the model.
`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"` or `"gradient"` is permitted.
`...`	When `type` is not `"wald"`, other optional arguments to be passed on the internal functions: - `lim` to specify the range of the evaluation points (default to (-3,3) resulting in the interval $\hat{\beta} \pm 3 \times \mathrm{SE}(\hat{\beta})$); - `values` to set explicitly the evaluation point(s); - `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`.

Details

Confidence intervals are computed and returned. Currently the Wald, Gradient or Likelihood ratio statistic can be used. Based on some simulation experiments the simple Wald based CIs appears adeguate to guarantee the nominal coverage levels.

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, lik=TRUE) #ML estimation

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 *kernel* 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. Currenty not used.
`k`	Optional numeric specifying the penalty of the edf in the AIC formula. If `k<=0`, the BIC is returned.
`...`	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, $\sum(\log(y_i)-\log(\hat\mu_i))^2)$. The value returned by `AIC` is based on the *kernel* log likelihood or the the sum of log residuals, while `extractAIC` can return the AIC (or BIC) using the full log likelihood (via `extractAIC(.., full=TRUE)`)

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, lik=TRUE) #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 = FALSE, opt = c("nlminb", "optim"), contrasts=NULL, ...)
```

Arguments

`formula`	a standard R formula with response and explanatory variables (and possible offset) 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 (positive) weights to be used in the fitting process. Currently implemented only if `lik=FALSE`.
`na.action`	a function indicating what should happen when the data contain NAs. The default is set by the `na.action` setting of `options`.
`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 $max(\hat{\beta}_0, median\{y_i\})$. If `lik=TRUE` (i.e. a log Normal model is fitted), `start` refers to the regression parameters *and* the error standard deviation; if `lik=FALSE`, `start` does *not* 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` appears to be more efficient, probably because it uses (unlike `optim`) also the hessian matrix (supplied in the code). However results are often indistinguishable.
`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 *consistent* 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

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

Author(s)

Vito M.R. Muggeo

See Also

See also `print.lognlm` and `summary.lognlm` to display results.

Examples

```
n=300
s=.4
set.seed(123)      #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, lik=TRUE) #fit the 'right' model by ML

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, weights=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 `lognlm`
`opt`	the optimizing algorithm. Default to `nlminb`.

`offset`	a possible offset term.
`weights`	apossible weights to be used if `lik=FALSE`.
`...`	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
```

`palermo` *Air quality in Palermo (Italy), 1997-2001*

Description

Daily time series of some pollutants and meteorological variables in Palermo, 1997-2001

Usage

```
data("palermo")
```

Format

A data frame with 1826 observations on the following 8 variables.

`day` day of month

`month` month of year

`year` year

`so2` Sulfur dioxide

`no2` Nitrogen dioxide

`pm10` particular matter

`temp` tempearture (degrees Celsius)

`hum` air humidity (%)

Details

Data refer to air pollution, temperature and humidity registered in Palermo, (Sicily, Italy) in 1997-2001. Data are averages from eight monitoring stations in the city.

Examples

```
## Not run:
data(palermo)

o<-lognlm(pm10 ~ temp + hum, data=palermo)



## End(Not run)
```

`paris` *PM2.5 and PM10 measurements in Paris in 2019*

Description

Hourly time series of PM2.5 and PM10 measurements in Paris in 2019

Usage

```
data("paris")
```

Format

A data frame with 647 observations on the following 4 variables.

`utc` date and time of measurements

`pm25` The PM2.5 measurements

`pm10` The PM10 measurements

`hours` numeric, the measurement hours

Details

Ambient particulate matter measurements (PM 2.5 and PM 10) measured by reference-grade instruments for Paris - Centre, hourly measurements for July 2019. Non-physical measurements (zero values and outliers with PM2.5 greater than 2xPM10) were removed, as mass of PM2.5 is a fraction of mass of PM10 by definition.

Source

European Environmental Agency, through Open AQ (`https://openaq.org/#/location/4146`). Thanks to Vito Ilacqua (EPA) for pointing out that.

Examples

```
data(paris)
plot(pm10~pm25, data=paris)
o<-lm(pm10~pm25, data=paris) #negative intercept! it's meaningless..
o1<-lognlm(pm10 ~ pm25, data=paris, lik=TRUE)
abline(coef=coef(o),  col=2)
abline(coef=coef(o1), col=3)
```

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

Description

Printing the most important feautures of a `'lognlm'` model.

Usage

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

Arguments

`x`	object of class segmented
`digits`	number of digits to be printed
`...`	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 `sandw=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` *Covariance matrix for lognlm fits*

Description

Computes covariance matrix of parameter estimates from a `lognlm` fit via the sandwich formula.

Usage

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

Arguments

`object`	a fitted model object of class `"lognlm"` returned by `lognlm()`
`emp`	logical; if `TRUE`, the 'meat' (i.e the information matrix) is computed empirically by the outer product of the individual score contributions.
`exH`	logical; if `TRUE` the *expected* (rather than the observed) hessian is used in the sandwich formula.
`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 log response standard deviation. Otherwise (if lik=FALSE has been set), it includes entries relevant to regression coefficients only. The var-covariance matrix comes from the sandwich formula using *expected* (if exH=TRUE) or the observed (if exH=FALSE) hessian at solution. Some simulations under correct model specification show that emp=TRUE and exH=FALSE lead to somewhat more unstable standard errors.

Value

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

Note

Currently for likelihood-based fits, exH=FALSE and emp=TRUE are always set.

Author(s)

Vito Muggeo

See Also

lognlm

Examples

```
n=50
s=.3

#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, lik=TRUE) #the model
vcov(o) #the full covariance matrix
vcov(o, se=TRUE) #st.errs only
```

Index