Package 'GLDreg'

February 28, 2017

Type Package

Title Fit GLD Regression Model and GLD Quantile Regression Model to Empirical Data

Version 1.0.7

Date 2017-03-15

Author Steve Su, with contributions from: R core team for qqgld.default function.

Maintainer Steve Su <allegro.su@gmail.com>

Depends GLDEX (>= 2.0.0.5), ddst, grDevices, graphics, stats

Suggests MASS, quantreg

Description

Owing to the rich shapes of Generalised Lambda Distributions (GLDs), GLD standard/quantile regression is a competitive flexible model compared to standard/quantile regression. The proposed method has some major advantages: 1) it provides a reference line which is very robust to outliers with the attractive property of zero mean residuals and 2) it gives a unified, elegant quantile regression model from the reference line with smooth regression coefficients across different quantiles. The goodness of fit of the proposed model can be assessed via QQ plots and Kolmogorov-Smirnov tests and data driven smooth test, to ensure the appropriateness of the statistical inference under consideration. Statistical distributions of coefficients of the GLD regression line are obtained using simulation, and interval estimates are obtained directly from simulated data.

License GPL (>= 3)

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-28 10:58:56

R topics documented:

`GLDreg-package` *This package fits standard and quantile regression models using RS and FMKL/FKML generalised lambda distributions via maximum likelihood estimation and L moment matching.*

Description

Owing to the rich shapes of GLDs, GLD standard/quantile regression is a competitive flexible model compared to standard/quantile regression. The proposed method has some major advantages: 1) it provides a reference line which is very robust to outliers with the attractive property of zero mean residuals and 2) it gives a unified, elegant quantile regression model from the reference line with smooth regression coefficients across different quantiles. The goodness of fit of the proposed model can be assessed via QQ plots and Kolmogorov-Smirnov tests and Data Driven Smooth Test, to ensure the appropriateness of the statistical inference under consideration. Statistical distributions of coefficients of the GLD regression line are obtained using simulation, and interval estimates are obtained directly from simulated data.

Details

Package:	GLDreg
Type:	Package
Version:	1.0.7
Date:	2017-03-15
License:	GPL (>= 3.0)

The primary fitting function for GLD regression model is `GLD.lm.full`. The output of `GLD.lm.full` can then be passed to `summaryGraphics.gld.lm` to display coefficients of GLD regression model graphically. Once a GLD reference model is obtained, quantile regression is obtained using `GLD.quantreg`.

Author(s)

Steve Su <allegro.su@gmail.com>

References

Su (2015) "Flexible Parametric Quantile Regression Model" Statistics & Computing May 2015, Volume 25, Issue 3, pp 635-650

See Also

GLDEX

Examples

```
## Dummy example

## Create dataset

set.seed(10)

x<-rnorm(200,3,2)
y<-3*x+rnorm(200)

dat<-data.frame(y,x)

## Fit a FKML GLD regression

example<-GLD.lm(y~x,data=dat,fun=fun.RMFMKL.ml.m,param="fkml")

## Fit FKML GLD regression with 3 simulations

fit<-GLD.lm.full(y~x,data=dat,fun=fun.RMFMKL.ml.m,param="fkml",n.simu=3)

## Find median regression, use empirical method

med.fit<-GLD.quantreg(0.5,fit,slope="fixed",emp=TRUE)

## Not run:

## Extract the Engel dataset

library(quantreg)
data(engel)

## Fit GLD Regression along with simulations

engel.fit.all<-GLD.lm.full(foodexp~income,data=engel,
param="fmkl",fun=fun.RMFMKL.ml.m)

## Plot coefficient summary

summaryGraphics.gld.lm(engel.fit.all)

## Fit quantile regression from 0.1 to 0.9, with equal spacings between
## quantiles

result<-GLD.quantreg(seq(0.1,.9,length=9),engel.fit.all,intercept="fixed")

## Plot quantile regression lines

fun.plot.q(x=engel$income,y=engel$foodexp,fit=engel.fit.all[[1]],result,
xlab="income",ylab="Food Expense")

## Sometimes the maximum likelihood estimation may fail, for example when
```

```
## minimum/maximum support of GLD is exactly at the minimum/maximum value of the
## dataset, if this the case, try to use the L-moment matching method.

engel.fit.all<-GLD.lm.full(foodexp~income,data=engel,
param="fmkl",fun=fun.RMFMKL.lm)


## End(Not run)
```

`fun.mean.convert` *Convert a RS or FKML GLD into RS or FKML GLD to the desired theoretical mean by changing only the first parameter*

Description

A simple transformation of altering the location of RS/FKML GLD so that the theoretical mean is altered to the level specified. Only the first parameter of RS/FKML GLD is altered.

Usage

```
fun.mean.convert(x, param, val = 0)
```

Arguments

`x`	A vector of four values representing Lambda 1, Lambda 2, Lambda 3 and Lambda 4 of RS/FKML GLD.
`param`	Can be "rs" or "fmkl" or "fkml"
`val`	The targeted theoretical mean

Value

A vector of four values representing Lambda 1, Lambda 2, Lambda 3 and Lambda 4 of the transformed RS/FKML GLD

Note

If finite first moment does not exist, original input values will be returned

Author(s)

Steve Su

Examples

```
# Transform RS GLD with parameters 3,2,1,1 to mean of 0
fun.mean.convert(c(3,2,1,1),param="rs")

# Check that the desired outcome is achieved
fun.theo.mv.gld(0,2,1,1,param="rs")

# Transform RS GLD with parameters 3,2,1,1 to mean of 5
fun.mean.convert(c(3,2,1,1),param="fkml",5)

# Check that the desired outcome is achieved
fun.theo.mv.gld(5,2,1,1,param="fkml")
```

`fun.plot.q` *2-D Plot for Quantile Regression lines*

Description

This function plots quantile regression lines from `GLD.lm` and one of `fun.gld.slope.vary.int.fixed`, `fun.gld.slope.fixed.int.vary`, `fun.gld.slope.fixed.int.vary.emp`, `fun.gld.all.vary.emp`, `fun.gld.all.vary`, `fun.gld.slope.vary.int.fixed.emp`, `GLD.quantreg`.

Usage

```
fun.plot.q(x, y, fit, quant.info, ...)
```

Arguments

`x`	A numerical vector of explanatory variable
`y`	A numerical vector of response variable
`fit`	An object from `GLD.lm`
`quant.info`	An object from one of `fun.gld.slope.vary.int.fixed`, `fun.gld.slope.fixed.int.vary`, `fun.gld.slope.fixed.int.vary.emp`, `fun.gld.all.vary.emp`, `fun.gld.all.vary`, `fun.gld.slope.vary.int.fixed.emp`, `GLD.quantreg`
`...`	Additional arguments to be passed to plot function, such as axis labels and title of the graph

Details

This is intended to plot only two variables, for quantile regression involving more than one explanatory variable, consider plotting the actual values versus fitted values by fitting a secondary GLD quantile model between actual and fitted values.

Value

A graph showing quantile regression lines

Author(s)

Steve Su

References

Su (2015) "Flexible Parametric Quantile Regression Model" Statistics & Computing May 2015, Volume 25, Issue 3, pp 635-650

Examples

```
## Dummy example

## Create dataset

set.seed(10)

x<-rnorm(200,3,2)
y<-3*x+rnorm(200)

dat<-data.frame(y,x)

## Fit FKML GLD regression with 3 simulations

fit<-GLD.lm.full(y~x,data=dat,fun=fun.RMFMKL.ml.m,param="fkml",n.simu=3)

## Find median regression, use empirical method

med.fit<-GLD.quantreg(0.5,fit,slope="fixed",emp=TRUE)

fun.plot.q(x=x,y=y,fit=fit[[1]],med.fit, xlab="x",ylab="y")

## Not run:

## Plot result of quantile regression

## Extract the Engel dataset

library(quantreg)
data(engel)

## Fit GLD Regression along with simulations

engel.fit.all<-GLD.lm.full(foodexp~income,data=engel,
param="fmkl",fun=fun.RMFMKL.ml.m)

## Fit quantile regression from 0.1 to 0.9, with equal spacings between
## quantiles

result<-GLD.quantreg(seq(0.1,.9,length=9),engel.fit.all,intercept="fixed")

## Plot the quantile regression lines
```

```
fun.plot.q(x=engel$income,y=engel$foodexp,fit=engel.fit.all[[1]],result,
xlab="income",ylab="Food Expense")

## End(Not run)
```

GLD.lm *This function fits a GLD regression linear model*

Description

Similar to lm, this function fits a linear model using RS/FKML GLDs and assess the goodness of fit of GLD with respect to the data via qq plot and Kolmogorov-Smirnoff (KS) test. Note that the use of KS test when parameters of a distribution are estimated from data is generally frowned upon. This is because one often gets inflated p-value with increased type II error due to the fact that the KS test requires independence between test sample and parameters of distribution. Therefore, the the resample KS test over 1000 simulation runs from GLDEX package is probably a more reasonable measure. It is probably reasonable to consider the resample KS test may in fact decrease the p-values, as testing is done on resampled data from fitted distribution so there is a certain degree of inaccuracy there. The provision of these results is to give some indication of optimistic and pessimistic goodness of fit measure, as currently, there is an absence of a specialised GLD goodness of fit test. A generic Data Driven Smooth Test from ddst library in R is also incorporated to assess goodness of fit.

When in doubt, QQ plot should always be considered ahead of these results.

Usage

```
GLD.lm(formula, data, param, maxit = 20000, fun, method = "Nelder-Mead",
diagnostics = TRUE, range = c(0.01, 0.99), init = NULL, alpha = 0.05)
```

Arguments

formula	A symbolic expression of the model to be fitted, similar to the formula argument in lm, see formula for more information
data	Dataset containing variables of the model
param	Can be "rs", "fmkl" or "fkml"
maxit	Maximum number of iterations for numerical optimisation
fun	If param="fmkl" or "fkml", this can be one of fun.RMFMKL.ml.m, fun.RMFMKL.ml, for maximum likelihood estimation (*.ml.m is a faster implementation of *.ml) and fun.RMFMKL.lm for L moment matching. If param="rs", this can be one of fun.RPRS.ml.m, fun.RPRS.ml, for maximum likelihood estimation (*.ml.m is a faster implementation of *.ml) and fun.RPRS.lm for L moment matching.
method	Defaults to "Nelder-Mead" algorithm, can also be "SANN" but this is a lot slower and may not as good

diagnostics	Defaults to TRUE, which computes Kolmogorov-Smirnoff test and do QQ plot
range	The is the quantile range to plot the QQ plot, defaults to 0.01 and 0.99 to avoid potential problems with extreme values of GLD which might be -Inf or Inf.
init	Choose a different set of initial values to start the optimisation process. This can either be full set of parameters including GLD parameter estimates, or it can just be the coefficient estimates of the regression model.
alpha	Significant level of KS test.

Value

Message	Short description of estimation method used and whether the result converged
Bias Correction	Bias correction used to ensure the line has zero mean residuals
Estimated parameters	A set of estimate coefficients from GLD regression
Fitted	Predicted response value from model
Residual	Residual of model
formula	Formula used in the model
param	Specify whether RS/FKML/FMKL GLD was used
y	The response variable
x	The explanatory variable(s)
fun	GLD fitting function used in the computation process, outputted for internal programming use

Author(s)

Steve Su

References

Su (2015) "Flexible Parametric Quantile Regression Model" Statistics & Computing May 2015, Volume 25, Issue 3, pp 635-650

See Also

GLD.lm.full, GLD.quantreg

Examples

```
## Dummy example

library(GLDEX)

## Create dataset

set.seed(10)
```

```
x<-rnorm(200,3,2)
y<-3*x+rnorm(200)

dat<-data.frame(y,x)

## Fit a FKML GLD regression

example<-GLD.lm(y~x,data=dat,fun=fun.RMFMKL.ml.m,param="fkml")

## Not run:

## Extract the Engel dataset
library(quantreg)
data(engel)

## Fit GLD Regression
engel.fit<-GLD.lm(foodexp~income,data=engel,param="fmkl",fun=fun.RMFMKL.ml.m)

## Extract the mammals dataset
library(MASS)

mammals.fit<-GLD.lm(log(brain)~log(body),data=mammals,param="rs",
fun=fun.RPRS.lm)

## Using quantile regression coefficients as starting values
library(quantreg)

mammals.fit1<-GLD.lm(log(brain)~log(body),data=mammals,param="rs",
fun=fun.RPRS.lm,init=rq(log(brain)~log(body),data=mammals)$coeff)

# As an exercise, use the result from mammals.fit1 as initial values

GLD.lm(log(brain)~log(body),data=mammals,param="rs",
fun=fun.RPRS.lm,init=mammals.fit1[[3]])

## End(Not run)
```

`GLD.lm.full` *This function fits a GLD regression linear model and conducts simulations to display the statistical properties of estimated coefficients*

Description

The function is an extension of `GLD.lm` and defaults to 1000 simulation runs, coefficients and statistical properties of coefficients can be plotted as part of the output.

Usage

```
GLD.lm.full(formula, data, param, maxit = 20000, fun, method = "Nelder-Mead",
range = c(0.01, 0.99), n.simu = 1000, summary.plot = TRUE, init = NULL)
```

Arguments

formula	A symbolic expression of the model to be fitted, similar to the formula argument in lm, see formula for more information
data	Dataset containing variables of the model
param	Can be "rs", "fmkl" or "fkml"
maxit	Maximum number of iterations for numerical optimisation
fun	If param="fmkl" or "fkml", this can be one of fun.RMFMKL.ml.m, fun.RMFMKL.ml, for maximum likelihood estimation (*.ml.m is a faster implementation of *.ml) and fun.RMFMKL.lm for L moment matching. If param="rs", this can be one of fun.RPRS.ml.m, fun.RPRS.ml, for maximum likelihood estimation (*.ml.m is a faster implementation of *.ml) and fun.RPRS.lm for L moment matching.
method	Defaults to "Nelder-Mead" algorithm, can also be "SANN" but this is a lot slower and may not as good
range	The is the quantile range to plot the QQ plot, defaults to 0.01 and 0.99 to avoid potential problems with extreme values of GLD which might be -Inf or Inf.
n.simu	Number of times to repeat the simulation runs, defaults to 1000.
summary.plot	Whether to plot the coefficients graphically, defaults to TRUE.
init	Choose a different set of initial values to start the optimisation process. This can either be full set of parameters including GLD parameter estimates, or it can just be the coefficient estimates of the regression model.

Details

This function usually takes some time to run, as it involves refitting the GLD regression model many times, the progress of the simulation is outputted to the R screen, so users can guage the progress of the computation.

Value

[[1]]	Output of GLD.lm
[[2]]	A matrix showing the bias adjustment, coefficents of the model, parameters of GLD and whether the result converged at each run
[[3]]	Adjusted simulation result so that the empirical mean of coefficients is the same as the estimated parameters obtained in GLD.lm

Author(s)

Steve Su

References

Su (2015) "Flexible Parametric Quantile Regression Model" Statistics & Computing May 2015, Volume 25, Issue 3, pp 635-650

See Also

GLD.lm, GLD.quantreg, summaryGraphics.gld.lm

Examples

```
## Dummy example

## Create dataset

set.seed(10)

x<-rnorm(200,3,2)
y<-3*x+rnorm(200)

dat<-data.frame(y,x)

## Fit FKML GLD regression with 3 simulations

fit<-GLD.lm.full(y~x,data=dat,fun=fun.RMFMKL.ml.m,param="fkml",n.simu=3)

## Not run:
## Extract the Engel dataset

library(quantreg)
data(engel)

## Fit a full GLD regression

engel.fit.full<-GLD.lm.full(foodexp~income,data=engel,param="fmkl",
fun=fun.RMFMKL.ml.m)

## Extract the mammals dataset
library(MASS)

## Fit a full GLD regression

mammals.fit.full<-GLD.lm.full(log(brain)~log(body),data=mammals,param="fmkl",
fun=fun.RMFMKL.ml.m)

## Using quantile regression coefficients as starting values
library(quantreg)

mammals.fit1.full<-GLD.lm.full(log(brain)~log(body),data=mammals,param="fmkl",
fun=fun.RMFMKL.ml.m, init=rq(log(brain)~log(body),data=mammals)$coeff)

## Using the result of mammals.fit.full as initial values

mammals.fit2.full<-GLD.lm.full(log(brain)~log(body),data=mammals,param="fmkl",
fun=fun.RMFMKL.ml.m, init=mammals.fit1.full[[1]][[3]])

## End(Not run)
```

GLD.quantreg *Fit a GLD quantile regression parametrically or non parametrically*

Description

The GLD quantile regression can be: 1) Fixed intercept, allowing all other coefficients to vary, 2) Only intercept is allowed to vary and 3) All coefficients can vary. Minimisation is achieved numerically through least squares between the proportion of estimated GLD error distribution below zero versus the specified quantile for parametric approach. For non parametric approach, minimisation is achieved using a least squares approach to find a q-th quantile GLD line such that the percentage of observations below the line corresponds to the q-th quantile.

Usage

```
GLD.quantreg(q, fit.obj, intercept = "", slope = "", emp=FALSE)
```

Arguments

q	Specify the quantile (range 0 to 1) line
fit.obj	An object from GLD.lm.full
intercept	Can either be "fixed" or left blank, blank indicates this parameter is allowed to vary in quantile line estimation
slope	Can either be "fixed" or left blank, blank indicates this parameter is allowed to vary in quantile line estimation
emp	Can either be TRUE (non parametric GLD quantile regression) or FALSE (parametric GLD quantile regression), defaults to FALSE

Details

This is a wrapper function for fun.gld.all.vary, fun.gld.slope.fixed.int.vary, fun.gld.slope.vary.int.fixed.

Value

A matrix showing the estimated coefficients for the specified quantile regression model, the objective function value and whether convergence is reached in the optimisation process. A value of 0 indicates convergence is reached. The convergence value is the same as the one from the optim function.

Author(s)

Steve Su

References

Su (2015) "Flexible Parametric Quantile Regression Model" Statistics & Computing May 2015, Volume 25, Issue 3, pp 635-650

See Also

GLD.lm.full,fun.plot.q, summaryGraphics.gld.lm

Examples

```
## Dummy example

## Create dataset

set.seed(10)

x<-rnorm(200,3,2)
y<-3*x+rnorm(200)

dat<-data.frame(y,x)

## Fit FKML GLD regression with 3 simulations

fit<-GLD.lm.full(y~x,data=dat,fun=fun.RMFMKL.ml.m,param="fkml",n.simu=3)

## Find median regression, use empirical method

med.fit<-GLD.quantreg(0.5,fit,slope="fixed",emp=TRUE)

## Not run:

## Extract the Engel dataset

library(quantreg)
data(engel)

## Fit GLD Regression along with simulations

engel.fit.all<-GLD.lm.full(foodexp~income,data=engel,
param="fmkl",fun=fun.RMFMKL.ml.m)

## Fit parametric GLD quantile regression from 0.1 to 0.9, with equal spacings
## between quantiles

result<-GLD.quantreg(seq(0.1,.9,length=9),engel.fit.all,intercept="fixed")

## Non parametric quantile regression

GLD.quantreg(seq(0.1,.9,length=9),engel.fit.all,intercept="fixed",emp=T)


## End(Not run)
```

qqgld.default *QQ plot for GLD*

Description

This is an updated QQ plot function for GLD comparing fitted distribution with empirical data

Usage

```
qqgld.default(y, vals, param, ylim, main = "GLD Q-Q Plot",
xlab = "Theoretical Quantiles", ylab = "Sample Quantiles",
plot.it = TRUE, datax = FALSE, ...)
```

Arguments

y	A vector of empirical data observations
vals	A vector representing four parameters of GLD
param	Can be "rs", "fmkl" or "fkml"
ylim	A vector of two numerical values, specifying the upper and lower bound of y axis
main	Title of the qq plot
xlab	Label for X axis
ylab	Label for Y axis
plot.it	Whether to plot the QQ plot, default is TRUE
datax	Whether data values should be on x axis, default is FALSE
...	Additional graphical parameters

Details

This is an adaptation of the default qq plot in R

Value

A list with components:

x	The x coordinates of the points that were/would be plotted
y	The original y vector, i.e., the corresponding y coordinates including NAs.

Author(s)

R, with modifications from Steve Su

See Also

qqplot.gld, qqplot.gld.bi

Examples

```
x<-rnorm(100)
fit1<-fun.RMFMKL.ml.m(x)
qqgld.default(x,fit1,param="fmkl")
```

---

summaryGraphics.gld.lm

Graphical display of output from GLD.lm.full

Description

This function display the coefficients and the distribution of coefficients obtained from GLD regression model. For a discussion on goodness of fit, please see the description under GLD.lm.

Usage

```
summaryGraphics.gld.lm(overall.fit.obj, alpha = 0.05, label = NULL,
ColourVersion = TRUE, diagnostics = TRUE, range = c(0.01, 0.99))
```

Arguments

overall.fit.obj	An object from GLD.lm.full
alpha	Specifying the range of interval for the coefficients, default is 0.05, which specifies a 95% interval. This also specifies the significance level of KS resample test.
label	A character vector indicating the labelling for the coefficients
ColourVersion	Whether to display colour or not, default is TRUE, if set as FALSE, a black and white plot is given. This is only applicable to the coefficient summary graph and has no effect on QQ plots.
diagnostics	If TRUE, then QQ plot will be given along with Kolmogorov-Smirnoff test results
range	The is the quantile range to plot the QQ plot, defaults to 0.01 and 0.99 to avoid potential problems with extreme values of GLD which might be -Inf or Inf.

Details

The reason QQ plots are not displayed in black and white even if ColourVersion is set to FALSE is because the colour is necessary in those plots for clarity of display.

Value

Graphics displaying coefficient summary and diagnostic plot (if chosen)

Author(s)

Steve Su

References

Su (2015) "Flexible Parametric Quantile Regression Model" Statistics & Computing May 2015, Volume 25, Issue 3, pp 635-650

See Also

GLD.lm.full

Examples

```
## Dummy example

## Create dataset

set.seed(10)

x<-rnorm(200,3,2)
y<-3*x+rnorm(200)

dat<-data.frame(y,x)

## Fit FKML GLD regression with 3 simulations

fit<-GLD.lm.full(y~x,data=dat,fun=fun.RMFMKL.ml.m,param="fkml",n.simu=3)

## Note this is for illustration only, need to set number
## of simulations around 1000 usually for the graphics below
## to be meaningful

summaryGraphics.gld.lm(fit,ColourVersion=FALSE,diagnostic=FALSE)

## Not run:
## Extract the Engel dataset

library(quantreg)
data(engel)

## Fit a full GLD regression

engel.fit.full<-GLD.lm.full(foodexp~income,data=engel,param="fmkl",
fun=fun.RMFMKL.ml.m)

## Plot coefficient summary

summaryGraphics.gld.lm(engel.fit.full,ColourVersion=FALSE,diagnostic=FALSE)

summaryGraphics.gld.lm(engel.fit.full)

## Extract the mammals dataset
library(MASS)
```

```
## Fit a full GLD regression

mammals.fit.full<-GLD.lm.full(log(brain)~log(body),data=mammals,param="fmkl",
fun=fun.RMFMKL.ml.m)

## Plot coefficient summary

summaryGraphics.gld.lm(mammals.fit.full,label=c("intercept","log of body weight"))


## End(Not run)
```

Index