Package ‘resplsm’

November 26, 2017

Type Package

Title Robust estimator for semi-parametric dynamic locationscale models

Version 0.0.0.9000

Author Edvinas Drevinskas

Maintainer Edvinas Drevinkskas <e.drevinskas@gmail.com>

Description Estimation of location scale parameters for stationary times series using robust semi-parametric method.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

R topics documented:

espls *Kernel M-Estimator for Location Scale model*

Description

Estimates parameters for location scale model using Kernel M-Estimator using R optim function

Usage

```
espls(Yt, St, s, initial.values, bandwidth = 1.06 * sqrt(var(St)) *
  length(St)^(-1/5), int.of.par = c(0, 1), print = F)
```

Arguments

Yt	parmeter of a function which is not to be optimized, usually Y_t
St	regresor parameter can be X's or lag(Y_t)
s	points at which function should be estimated
initial.values	initial value of optimisible parameter might be a vector
bandwidth	bandwith should be used
int.of.par	initial parameters
print	print during fitting

Value

Estimated location scale function at s points

interpolate_theta *Get $\theta(X)$*

Description

Get $\theta(X)$

Usage

```
interpolate_theta(dat, X)
```

Arguments

dat	data.frame which contains X and value of thate for that X
X	vector for which new values of $\theta(X)$ should be returned

Value

Returns $\theta(X)$

k1 *k1*

Description

$$k_1 = \frac{1}{2v_0(y_{t-1})^2}\frac{\partial v_0(y_{t-1})^2}{\partial\theta(y_{t-1})}\Big|_{\theta=\theta_0}$$

Usage

```
k1(theta, x, func)
```

Arguments

`theta`	A data.frame
`x`	A number/vectors.
`func`	scale function

Value

A value of score function.

k2 *k2*

Description

$$k_2 = \frac{1}{v_0(y_{t-1})}\frac{\partial m_0(y_{t-1})^2}{\partial\theta(y_{t-1})}\Big|_{\theta=\theta_0}$$

Usage

```
k2(theta, x, func)
```

Arguments

`theta`	A data.frame
`x`	A number/vectors.
`func`	scale function

Value

A value of score function.

`Laplace_approx` *Laplace_approx*

Description

$\mathcal{L}(\overline{\mathbf{q}_n}, \overline{u})$ for the τ numerator

Usage

```
Laplace_approx(u, parameters, h = 1e-04)
```

Arguments

`u`	A number
`parameters`	A list with given parameters to function: `k1m`, `k2m`, `A`, `cb`, `tau`, `func_mu`, `func_sigma`, `x`, `theta0`
`h`	numerical derivative parameter

Value

value of Laplace_approx function

`Laplace_approx_den` *Laplace_approx_den*

Description

$\mathcal{L}(\overline{\mathbf{q}_n}, \overline{u})$ for the τ denominator

Usage

```
Laplace_approx_den(u, parameters, h = 1e-04)
```

Arguments

`u`	A number
`parameters`	A list with given parameters to function: `k1m`, `k2m`, `A`, `cb`, `tau`, `func_mu`, `func_sigma`, `x`, `theta0`
`h`	numerical derivative parameter

Value

value of Laplace_approx_den function

pls *Pseudo Liklyhood Estimator for Location Scale model*

Description

Description

Usage

```
pls(initial.theta, Y, X, func_s, func_m)
```

Arguments

`initial.theta`	initial value of theta, A vector
`Y`	parmeter of a function which is not to be optimized, usually Y_t
`X`	regresor parameter can be X's or lag(Y_t)
`func_s`	scale function
`func_m`	location function

Value

Estimated location theta Robust

qn_function *qn_function*

Description

$$q_n(u) := (-k_1 + k_2 u + k_1 u^2) \frac{c}{\|A(s(v;\theta_0) - \tau^{(0)})\|}$$

Usage

```
qn_function(u, parameters = list())
```

Arguments

`u`	A number
`parameters`	A list with given parameters to function: `k1m`, `k2m`, `A`, `cb`, `tau`, `func_mu`, `func_sigma`, `x`, `theta0`

Value

value of q_n function

qn_function_den *qn_function_den*

Description

$$q_n^{den}(u) := \frac{c}{\|A(s(v;\theta_0) - \tau^{(0)})\|}$$

Part of τ calculation

Usage

```
qn_function_den(u, parameters = list())
```

Arguments

u	A number
parameters	A list with given parameters to function: k1m, k2m, A, cb, tau, func_mu, func_sigma, x, theta0

Value

value of q_n function

qn_function_z *qn_function_z*

Description

$q_n(z) := q_n(u+z)\exp(-.5z^2)$

Usage

```
qn_function_z(z, u, parameters)
```

Arguments

z	A number
u	A number
parameters	A list with given parameters to function: k1m, k2m, A, cb, tau, func_mu, func_sigma, x, theta0

Value

value of q_n function

`qn_function_z_den` *qn_function_z_den*

Description

$q_n^{den}(z) := q_n^{den}(u + z)\exp(-.5z^2)$

Usage

```
qn_function_z_den(z, u, parameters)
```

Arguments

`z`	A number
`u`	A number
`parameters`	A list with given parameters to function: `k1m`, `k2m`, `A`, `cb`, `tau`, `func_mu`, `func_sigma`, `x`, `theta0`

Value

value of q_n function

`respls` *Robust Kernel M-Estimator for Location Scale model*

Description

Description

Usage

```
respls(theta, Y, X, c_bound, iterations = 5, bindwidths, return.all = F)
```

Arguments

`theta`	initial value of theta, document later
`Y`	parmeter of a function which is not to be optimized, usually Y_t
`X`	regresor parameter can be X's or lag(Y_t)
`c_bound`	bounding constant
`iterations`	number of iterantions
`bindwidths`	bindwidths should be used
`return.all`	if TRUE returns list of all $\theta^{(j)}$

Value

Estimated location theta Robust

rls *Robust M-Estimator for Location Scale model*

Description

Description

Usage

```
rls(theta, Y, X, c_bound, func_s, d_func_s, func_m, d_func_m, iterations = 5,
  return.all = F, tolerance = 0)
```

Arguments

theta	initial value of theta, document later
Y	parmeter of a function which is not to be optimized, usually Y_t
X	regresor parameter can be X's or lag(Y_t)
c_bound	bounding constant
func_s	scale function
d_func_s	derivative of scale function
func_m	location function
d_func_m	derivative of location function
iterations	number of iterantions
return.all	if TRUE returns list of all $\theta^{(j)}$
tolerence	tolerance level

Value

Estimated location theta Robust

semi_est_func *Estimating function*

Description

Estimating function

Usage

```
semi_est_func(yt, thetas)
```

Arguments

yt	A number.
thetas	A vector of lengths 2.

Value

A value of score function.

u_resids *Residuals*

Description

Residuals

Usage

```
u_resids(y, x, theta, func_mu, func_sigma)
```

Arguments

y	A number.
x	A number/vectors.
theta	A vector of lengths 2 of data.frame, depends on func_mu and func_sigma.
func_mu	location function
func_sigma	scale function

Value

residual

Index