

# Package ‘RobRex’

November 4, 2009

**Version** 0.7

**Date** 2009-10-16

**Title** Optimally robust influence curves for regression and scale

**Description** functions for the determination of optimally robust influence curves in case of linear regression with unknown scale and standard normal distributed errors where the regressor is random.

**Depends** R (>= 2.4.0), ROptRegTS(>= 0.6.1)

**Author** Matthias Kohl <Matthias.Kohl@stamats.de>

**Maintainer** Matthias Kohl <Matthias.Kohl@stamats.de>

**LazyLoad** yes

**License** LGPL-3

**Encoding** latin1

**URL** <http://robast.r-forge.r-project.org/>

**LastChangedDate** {\$LastChangedDate: 2009-10-16 19:39:14 +0200 (Fr, 16. Okt 2009) \$}

**LastChangedRevision** {\$LastChangedRevision: 391 \$}

**Repository** CRAN

**Date/Publication** 2009-11-04 13:09:44

## R topics documented:

rgsOptIC.AL	2
rgsOptIC.ALc	3
rgsOptIC.ALs	4
rgsOptIC.BM	6
rgsOptIC.M	7
rgsOptIC.Mc	8
rgsOptIC.MK	10
rgsOptIC.Ms	11

---

 rgsOptIC.AL

*Computation of the optimally robust IC for AL estimators*


---

### Description

The function `rgsOptIC.AL` computes the optimally robust IC for AL estimators in case of linear regression with unknown scale and (convex) contamination neighborhoods where the regressor is random; confer Subsubsection 7.2.1.1 of Kohl (2005).

### Usage

```
rgsOptIC.AL(r, K, theta, scale = 1, A.rg.start, a.sc.start = 0, A.sc.start = 0.5,
            bUp = 1000, delta = 1e-06, itmax = 50, check = FALSE)
```

### Arguments

<code>r</code>	non-negative real: neighborhood radius.
<code>K</code>	object of class "Distribution".
<code>theta</code>	specified regression parameter.
<code>scale</code>	specified error scale.
<code>A.rg.start</code>	positive definite and symmetric matrix: starting value for the standardizing matrix of the regression part.
<code>a.sc.start</code>	real: starting value for centering constant of the scale part.
<code>A.sc.start</code>	positive real: starting value for the standardizing constant of the scale part.
<code>bUp</code>	positive real: the upper end point of the interval to be searched for b.
<code>delta</code>	the desired accuracy (convergence tolerance).
<code>itmax</code>	the maximum number of iterations.
<code>check</code>	logical. Should constraints be checked.

### Details

If `theta` is missing, it is set to 0. If `A.rg.start` is missing, the inverse of the second moment matrix of `K` is used. The Lagrange multipliers contained in the expression of the optimally robust IC can be accessed via the accessor functions `cent`, `clip` and `stand`.

### Value

Object of class "ContIC"

### Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

**References**

- Rieder, H. (1994) *Robust Asymptotic Statistics*. New York: Springer.
- Kohl, M. (2005) *Numerical Contributions to the Asymptotic Theory of Robustness*. Bayreuth: Dissertation.

**See Also**

ContIC-class

**Examples**

```
K <- DiscreteDistribution(1:5) # = Unif({1,2,3,4,5})
IC1 <- rgsOptIC.AL(r = 0.1, K = K)
checkIC(IC1)
Risks(IC1)
cent(IC1)
clip(IC1)
stand(IC1)
```

---

 rgsOptIC.ALc

---

*Computation of the optimally robust IC for AL estimators*


---

**Description**

The function `rgsOptIC.ALc` computes the optimally robust conditionally centered IC for AL estimators in case of linear regression with unknown scale and average conditional (convex) contamination neighborhoods where the regressor is random; confer Subsubsection 7.2.1.2 of Kohl (2005).

**Usage**

```
rgsOptIC.ALc(r, K, theta, scale = 1, A.rg.start, a.sc.start, A.sc.start = 0.5,
             bUp = 1000, delta = 1e-06, itmax = 50, check = FALSE)
```

**Arguments**

<code>r</code>	non-negative real: neighborhood radius.
<code>K</code>	object of class "DiscreteDistribution" or object of class "DiscreteMVDistribution".
<code>theta</code>	specified regression parameter.
<code>scale</code>	specified error scale.
<code>A.rg.start</code>	positive definite and symmetric matrix: starting value for the standardizing matrix of the regression part.
<code>a.sc.start</code>	real vector: starting values for centering function of the scale part.
<code>A.sc.start</code>	positive real: starting value for the standardizing constant of the scale part.
<code>bUp</code>	positive real: the upper end point of the interval to be searched for b.

delta	the desired accuracy (convergence tolerance).
itmax	the maximum number of iterations.
check	logical. Should constraints be checked.

### Details

If `theta` is missing, it is set to 0. If `A.rg.start` is missing, the inverse of the second moment matrix of `K` is used. In case `a.sc.start` is missing, it is set to a null vector with length of the support of `K`.

### Value

Object of class "Av1CondContIC"

### Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

### References

Rieder, H. (1994) *Robust Asymptotic Statistics*. New York: Springer.  
 Kohl, M. (2005) *Numerical Contributions to the Asymptotic Theory of Robustness*. Bayreuth: Dissertation.

### See Also

Av1CondContIC-class

### Examples

```
K <- DiscreteDistribution(1:5) # = Unif({1,2,3,4,5})
IC1 <- rgsOptIC.ALc(r = 0.1, K = K)
checkIC(IC1)
Risks(IC1)
```

---

rgsOptIC.ALs

*Computation of the optimally robust IC for ALs estimators*

---

### Description

The function `rgsOptIC.ALs` computes the optimally robust IC for ALs estimators in case of linear regression with unknown scale and (convex) contamination neighborhoods where the regressor is random; confer Subsection 7.3.1 of Kohl (2005).

### Usage

```
rgsOptIC.ALs(r, K, A.rg.start, b.rg.Up = 1000, delta = 1e-06,
            itmax = 50, check = FALSE)
```

**Arguments**

<code>r</code>	non-negative real: neighborhood radius.
<code>K</code>	object of class "Distribution".
<code>A.rg.start</code>	positive definite and symmetric matrix: starting value for the standardizing matrix of the regression part.
<code>b.rg.Up</code>	positive real: the upper end point of the interval to be searched for <code>b.rg</code> .
<code>delta</code>	the desired accuracy (convergence tolerance).
<code>itmax</code>	the maximum number of iterations.
<code>check</code>	logical. Should constraints be checked.

**Details**

If `A.rg.start` is missing, the inverse of the second moment matrix of `K` is used.

**Value**

Object of class "ContIC"

**Author(s)**

Matthias Kohl <Matthias.Kohl@stamats.de>

**References**

Rieder, H. (1994) *Robust Asymptotic Statistics*. New York: Springer.

Kohl, M. (2005) *Numerical Contributions to the Asymptotic Theory of Robustness*. Bayreuth: Dissertation.

**See Also**

`ContIC-class`

**Examples**

```
## code takes some time
## Not run:
K <- DiscreteDistribution(1:5) # = Unif({1,2,3,4,5})
IC1 <- rgsOptIC.ALs(r = 0.1, K = K)
checkIC(IC1)
Risks(IC1)
Infos(IC1)

## End(Not run)
```

---

 rgsOptIC.BM

*Computation of the optimally robust IC for BM estimators*


---

### Description

The function `rgsOptIC.BM` computes the optimally robust IC for BM estimators in case of linear regression with unknown scale and (convex) contamination neighborhoods where the regressor is random. These estimators were proposed by Bednarski and Mueller (2001); confer also Subsection 7.3.3 of Kohl (2005).

### Usage

```
rgsOptIC.BM(r, K, b.rg.start = 2.5, b.sc.0.x.start, delta = 1e-06,
            MAX = 100, itmax = 1000)
```

### Arguments

<code>r</code>	non-negative real: neighborhood radius.
<code>K</code>	object of class "DiscreteDistribution"
<code>b.rg.start</code>	positive real: starting value for $b_{rg}$ .
<code>b.sc.0.x.start</code>	positive real: starting value for $b_{sc,0,x}$ .
<code>delta</code>	the desired accuracy (convergence tolerance).
<code>itmax</code>	the maximum number of iterations.
<code>MAX</code>	if $b_{loc}$ or $b_{sc,0}$ are beyond the admitted values, <code>MAX</code> is returned.

### Details

The computation of the optimally robust IC for BM estimators is based on `optim` where `MAX` is used to control the constraints on  $b_{rg}$  and  $b_{sc,0,x}$ .

### Value

Object of class "CondIC"

### Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

### References

Bednarski, T and Mueller, C.H. (2001) Optimal bounded influence regression and scale M-estimators in the context of experimental design. *Statistics*, **35**(4): 349–369.

Kohl, M. (2005) *Numerical Contributions to the Asymptotic Theory of Robustness*. Bayreuth: Dissertation.

**See Also**

CondIC-class

**Examples**

```
## code takes some time
## Not run:
K <- DiscreteDistribution(1:5) # = Unif({1,2,3,4,5})
IC1 <- rgsOptIC.BM(r = 0.1, K = K)
checkIC(IC1)
Risks(IC1)

## End(Not run)
```

---

 rgsOptIC.M

*Computation of the optimally robust IC for M estimators*


---

**Description**

The function `rgsOptIC.M` computes the optimally robust IC for M estimators in case of linear regression with unknown scale and (convex) contamination neighborhoods where the regressor is random; confer Subsubsection 7.2.2.1 of Kohl (2005).

**Usage**

```
rgsOptIC.M(r, K, A.start, gg.start = 0.6, a1.start = -0.25,
           a3.start = 0.25, B.start, bUp = 1000, delta = 1e-05,
           MAX = 100, itmax = 1000, check = FALSE)
```

**Arguments**

<code>r</code>	non-negative real: neighborhood radius.
<code>K</code>	object of class "Distribution".
<code>A.start</code>	positive definite and symmetric matrix: starting value for the standardizing matrix of the regression part.
<code>gg.start</code>	positive real: starting value for the standardizing constant $\gamma$ of the scale part.
<code>a1.start</code>	real: starting value for Lagrange multiplier $\alpha_1$ .
<code>a3.start</code>	real: starting value for Lagrange multiplier $\alpha_3$ .
<code>B.start</code>	symmetric matrix: starting value for Lagrange multiplier B.
<code>bUp</code>	positive real: the upper end point of the interval to be searched for b.
<code>delta</code>	the desired accuracy (convergence tolerance).
<code>MAX</code>	if A or $\gamma$ are beyond the admitted values, MAX is returned.
<code>itmax</code>	the maximum number of iterations.
<code>check</code>	logical. Should constraints be checked.

**Details**

The computation of the optimally robust IC for M estimators is based on `optim` where `MAX` is used to control the constraints on  $A$  and  $\gamma$ .

**Value**

Object of class "IC"

**Author(s)**

Matthias Kohl <Matthias.Kohl@stamats.de>

**References**

Kohl, M. (2005) *Numerical Contributions to the Asymptotic Theory of Robustness*. Bayreuth: Dissertation.

**See Also**

IC-class

**Examples**

```
## code takes some time
## Not run:
K <- DiscreteDistribution(1:5) # = Unif({1,2,3,4,5})
IC1 <- rgsOptIC.M(r = 0.1, K = K)
checkIC(IC1)
Risks(IC1)

## End(Not run)
```

---

rgsOptIC.Mc

*Computation of the optimally robust IC for Mc estimators*

---

**Description**

The function `rgsOptIC.Mc` computes the optimally robust conditionally centered IC for Mc estimators in case of linear regression with unknown scale and average conditional (convex) contamination neighborhoods where the regressor is random; confer Subsubsection 7.2.2.2 of Kohl (2005).

**Usage**

```
rgsOptIC.Mc(r, K, ggLo = 0.5, ggUp = 1, a1.x.start, a3.start = 0.25,
            bUp = 1000, delta = 1e-05, itmax = 1000, check = FALSE)
```

**Arguments**

<code>r</code>	non-negative real: neighborhood radius.
<code>K</code>	object of class "DiscreteDistribution"
<code>ggLo</code>	positive real: the lower end point of the interval to be searched for $\gamma$ .
<code>ggUp</code>	positive real: the upper end point of the interval to be searched for $\gamma$ .
<code>a1.x.start</code>	real: starting value for the Lagrange multiplier function $\alpha_1(x)$ .
<code>a3.start</code>	real: starting value for Lagrange multiplier $\alpha_3$ .
<code>bUp</code>	positive real: the upper end point of the interval to be searched for $b$ .
<code>delta</code>	the desired accuracy (convergence tolerance).
<code>itmax</code>	the maximum number of iterations.
<code>check</code>	logical. Should constraints be checked.

**Value**

Object of class "CondIC"

**Author(s)**

Matthias Kohl <Matthias.Kohl@stamats.de>

**References**

Kohl, M. (2005) *Numerical Contributions to the Asymptotic Theory of Robustness*. Bayreuth: Dissertation.

**See Also**

CondIC-class

**Examples**

```
## code takes some time
## Not run:
K <- DiscreteDistribution(1:5) # = Unif({1,2,3,4,5})
IC1 <- rgsOptIC.Mc(r = 0.1, K = K)
checkIC(IC1)
Risks(IC1)

## End(Not run)
```

---

 rgsOptIC.MK

*Computation of the optimally robust IC for MK estimators*


---

### Description

The function `rgsOptIC.MK` computes the optimally robust IC for MK estimators in case of linear regression with unknown scale and (convex) contamination neighborhoods where the regressor is random; confer Subsubsection 7.2.2.1 of Kohl (2005).

### Usage

```
rgsOptIC.MK(r, K, ggLo = 0.5, ggUp = 1, a1.start = -0.25, a3.start = 0.25,
            B.start, bUp = 1000, delta = 1e-06, itmax = 1000, check = FALSE)
```

### Arguments

<code>r</code>	non-negative real: neighborhood radius.
<code>K</code>	object of class "Distribution".
<code>ggLo</code>	positive real: the lower end point of the interval to be searched for $\gamma$ .
<code>ggUp</code>	positive real: the upper end point of the interval to be searched for $\gamma$ .
<code>a1.start</code>	real: starting value for Lagrange multiplier $\alpha_1$ .
<code>a3.start</code>	real: starting value for Lagrange multiplier $\alpha_3$ .
<code>B.start</code>	symmetric matrix: starting value for Lagrange multiplier B.
<code>bUp</code>	positive real: the upper end point of the interval to be searched for b.
<code>delta</code>	the desired accuracy (convergence tolerance).
<code>itmax</code>	the maximum number of iterations.
<code>check</code>	logical. Should constraints be checked.

### Value

Object of class "IC"

### Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

### References

Kohl, M. (2005) *Numerical Contributions to the Asymptotic Theory of Robustness*. Bayreuth: Dissertation.

### See Also

IC-class

**Examples**

```
## code takes some time
## Not run:
K <- DiscreteDistribution(1:5) # = Unif({1,2,3,4,5})
IC1 <- rgsOptIC.MK(r = 0.1, K = K)
checkIC(IC1)
Risks(IC1)

## End(Not run)
```

rgsOptIC.Ms

*Computation of the optimally robust IC for Ms estimators***Description**

The function `rgsOptIC.Ms` computes the optimally robust conditionally centered IC for Ms estimators in case of linear regression with unknown scale and average conditional (convex) contamination neighborhoods where the regressor is random; confer Subsection 7.3.2 of Kohl (2005).

**Usage**

```
rgsOptIC.Ms(r, K, a1.x.start, a3.start = 0.25, b.sc.start = 1.5,
            bUp = 1000, ggLo = 0.5, ggUp = 1, delta = 1e-06,
            itmax = 1000, check = FALSE)
```

**Arguments**

<code>r</code>	non-negative real: neighborhood radius.
<code>K</code>	object of class "DiscreteDistribution"
<code>ggLo</code>	positive real: the lower end point of the interval to be searched for $\gamma$ .
<code>ggUp</code>	positive real: the upper end point of the interval to be searched for $\gamma$ .
<code>a1.x.start</code>	real: starting value for the Lagrange multiplier function $\alpha_1(x)$ .
<code>a3.start</code>	real: starting value for Lagrange multiplier $\alpha_3$ .
<code>b.sc.start</code>	positive real: starting value for the clipping bound $b_{sc}$ .
<code>bUp</code>	positive real: the upper end point of the interval to be searched for $b$ .
<code>delta</code>	the desired accuracy (convergence tolerance).
<code>itmax</code>	the maximum number of iterations.
<code>check</code>	logical. Should constraints be checked.

**Value**

Object of class "CondIC"

**Author(s)**

Matthias Kohl <Matthias.Kohl@stamats.de>

**References**

Kohl, M. (2005) *Numerical Contributions to the Asymptotic Theory of Robustness*. Bayreuth: Dissertation.

**See Also**

CondIC-class

**Examples**

```
## code takes some time
## Not run:
K <- DiscreteDistribution(1:5) # = Unif({1,2,3,4,5})
IC1 <- rgsOptIC.Ms(r = 0.1, K = K)
checkIC(IC1)
Risks(IC1)

## End(Not run)
```

# Index

## \*Topic **robust**

- rgsOptIC.AL, [2](#)
- rgsOptIC.ALc, [3](#)
- rgsOptIC.ALs, [4](#)
- rgsOptIC.BM, [6](#)
- rgsOptIC.M, [7](#)
- rgsOptIC.Mc, [8](#)
- rgsOptIC.MK, [10](#)
- rgsOptIC.Ms, [11](#)

- rgsOptIC.AL, [2](#)
- rgsOptIC.ALc, [3](#)
- rgsOptIC.ALs, [4](#)
- rgsOptIC.BM, [6](#)
- rgsOptIC.M, [7](#)
- rgsOptIC.Mc, [8](#)
- rgsOptIC.MK, [10](#)
- rgsOptIC.Ms, [11](#)