Package 'robustreg'

August 26, 2015

Version 0.1-9

Date 2015-08-25

Title Robust Regression Functions

Author Ian M. Johnson <ian@alpha-analysis.com>

Maintainer Ian M. Johnson <ian@alpha-analysis.com>

Depends R (>= 3.0.0)

Description Linear regression functions using Huber and bisquare psi functions. Optimal weights are calculated using IRLS algorithm.

License GPL (>= 2)

Imports stats (>= 3.0.0), Matrix (>= 1.1.0), Rcpp (>= 0.11.3)

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-08-26 08:51:12

R topics documented:

fit_rcpp *Predict y from X and b*

Description

Predict y vector from X design matrix and b vector

Usage

```
fit_rcpp(X,b)
```

Arguments

X	Design matrix
b	Estimates of beta

Examples

```
## Not run:
fit_rcpp(X,b)

## End(Not run)
```

mad_rcpp *Median Absolute Deviation (MAD)*

Description

Rcpp fast implementation of median absolute deviation (MAD)

Usage

```
mad_rcpp(r,scale_factor = 1.4826)
```

Arguments

r	A numeric vector
scale_factor	Scale factor

Examples

```
## Not run:
mad_rcpp(r,scale_factor = 1.4826)

## End(Not run)
```

median_rcpp *Median*

Description

Rcpp fast implementation of median

Usage

```
median_rcpp(x)
```

Arguments

x	A numeric vector containing the values whose median is to be computed.

Examples

```
## Not run:
median_rcpp(x)

## End(Not run)
```

psiBS_rcpp *Tukey's Bisquare Psi Function*

Description

Rcpp fast implementation of Tukey's Bisquare psi function

Usage

```
psiBS_rcpp(r,c)
```

Arguments

r	A numeric vector
c	Tuning constant

Examples

```
## Not run:
psiBS_rcpp(r,c)

## End(Not run)
```

psiHuber_rcpp *Huber Psi Function*

Description

Rcpp fast implementation of Huber's Psi Function

Usage

```
psiHuber_rcpp(r,c)
```

Arguments

r	A numeric vector
c	Tuning constant

Examples

```
## Not run:
psiHuber_rcpp(r,c)

## End(Not run)
```

robustRegBS *Robust Fitting of Linear Models using Bisquare Psi Function*

Description

Using iteratively reweighted least squares (IRLS), the function calculates the optimal weights to perform m-estimator or bounded influence regression. Returns robust beta estimates and prints robust ANOVA table.

Usage

```
robustRegBS(formula,data,tune=4.685,m=TRUE,max.it=1000,tol=1e-5,anova.table=FALSE)
```

Arguments

formula	Model
data	A data frame containing the variables in the model.
tune	Tuning Constant. Default value of 4.685 is 95% asymptotically efficient against outliers
m	If TRUE, calculates m estimates of beta. If FALSE, calculates bounded influence estimates of beta
max.it	Maximum number of iterations to achieve convergence in IRLS algorithm
tol	Tolerance level in determining convergence
anova.table	If TRUE, prints robust ANOVA table

Details

M-estimates of beta should be used when evaluating least squares estimates of beta and diagnostics show outliers. Least squares estimates of beta should be used as starting points to achieve convergence.

Bounded influence estimates of beta should be used when evaluating least squares estimates of beta and diagnostics show large values of the "Hat Matrix" diagonals and outliers.

Note

Original package written in 2006

Author(s)

Ian M. Johnson <ian@alpha-analysis.com>

References

Tukey,

Birch, Robust F-Test, 1983

See Also

robustRegH()

Examples

```
data(stackloss)
robustRegBS(stack.loss~Air.Flow+Water.Temp,data=stackloss)

#If X matrix contained large values of H matrix (high influence points)
robustRegBS(stack.loss~Air.Flow+Water.Temp,data=stackloss,m=FALSE)
```

---

robustRegH *Robust Fitting of Linear Models using Huber Psi Function*

Description

Using iteratively reweighted least squares (IRLS), the function calculates the optimal weights to perform m-estimator or bounded influence regression. Returns robust beta estimates and prints robust ANOVA table

Usage

```
robustRegH(formula,data,tune=1.345,m=TRUE,max.it=1000,tol=1e-5,anova.table=FALSE)
```

Arguments

formula	Model
data	A data frame containing the variables in the model.
tune	Tuning Constant. Default value of 1.345 is 95% asymptotically efficient against outliers
m	If TRUE, calculates m estimates of beta. If FALSE, calculates bounded influence estimates of beta
max.it	Maximum number of iterations to achieve convergence in IRLS algorithm
tol	Tolerance level in determining convergence
anova.table	If TRUE, prints robust ANOVA table

Details

M-estimates of beta should be used when evaluating least squares estimates of beta and diagnostics show outliers. Least squares estimates of beta are used as starting points to achieve convergence.

Bounded influence estimates of beta should be used when evaluating least squares estimates of beta and diagnostics show large values of the "Hat Matrix" diagonals and outliers.

Note

Original package written in 2006

Author(s)

Ian M. Johnson <ian@alpha-analysis.com>

References

P. J. Huber (1981) Robust Statistics. Wiley.

Birch (1983) Robust F-Test

See Also

robustRegBS()

Examples

```
data(stackloss)
robustRegH(stack.loss~Air.Flow+Water.Temp,data=stackloss)

#If X matrix contained large values of H matrix (high influence points)
robustRegH(stack.loss~Air.Flow+Water.Temp,data=stackloss,m=FALSE)
```

Index