

# Package: RobMixReg (via r-universe)

August 23, 2024

**Type** Package

**Title** Robust Mixture Regression

**Version** 1.1.0

**Date** 2020-08-03

**Author** Sha Cao [aut, cph, ths], Wennan Chang [aut, cre], Chi Zhang [aut, ctb, ths]

**Maintainer** Wennan Chang <wnchang@iu.edu>

**Description** Finite mixture models are a popular technique for modelling unobserved heterogeneity or to approximate general distribution functions in a semi-parametric way. They are used in a lot of different areas such as astronomy, biology, economics, marketing or medicine. This package is the implementation of popular robust mixture regression methods based on different algorithms including: fleximix, finite mixture models and latent class regression; CTLERob, component-wise adaptive trimming likelihood estimation; mixbi, bi-square estimation; mixL, Laplacian distribution; mixt, t-distribution; TLE, trimmed likelihood estimation. The implemented algorithms includes: CTLERob stands for Component-wise adaptive Trimming Likelihood Estimation based mixture regression; mixbi stands for mixture regression based on bi-square estimation; mixLstands for mixture regression based on Laplacian distribution; TLE stands for Trimmed Likelihood Estimation based mixture regression. For more detail of the algorithms, please refer to below references. Reference: Chun Yu, Weixin Yao, Kun Chen (2017) <doi:10.1002/cjs.11310>. NeyKov N, Filzmoser P, Dimova R et al. (2007) <doi:10.1016/j.csda.2006.12.024>. Bai X, Yao W, Boyer JE (2012) <doi:10.1016/j.csda.2012.01.016>. Wennan Chang, Xinyu Zhou, Yong Zang, Chi Zhang, Sha Cao (2020) <arXiv:2005.11599>.

**Depends** R (>= 3.5.0)

**License** GPL

**Encoding** UTF-8

**LazyData** true

**Imports** flexmix, robustbase, gtools, MASS,  
 methods,robust,lars,dplyr,rlang,  
 scales,gplots,grDevices,graphics,RColorBrewer,stats,glmnet

**RoxygenNote** 6.1.1

**URL** <https://changwn.github.io/RobMixReg/>

**BugReports** <https://github.com/changwn/RobMixReg/issues>

**Repository** <https://changwn.r-universe.dev>

**RemoteUrl** <https://github.com/changwn/robmixreg>

**RemoteRef** HEAD

**RemoteSha** 4a74afca99f35dc26291c2f1cad20d78da2b6b65

## Contents

biscalew . . . . .	3
bisquare . . . . .	3
blockMap . . . . .	4
CCLE_data . . . . .	4
colon_data . . . . .	5
compPlot . . . . .	5
Compute_Rbase_SVD . . . . .	6
CSMR . . . . .	6
CSMR_one . . . . .	7
CSMR_predict . . . . .	7
CSMR_train . . . . .	8
CTLERob . . . . .	8
denLp . . . . .	9
DeOut . . . . .	9
flexmix_2 . . . . .	10
gaussData . . . . .	10
lars.lsa . . . . .	11
logLik_mixtureReg . . . . .	11
lsa . . . . .	12
mixlinrb_bi . . . . .	12
mixlinrb_bione . . . . .	13
mixLp . . . . .	14
mixLp_one . . . . .	15
mixtureReg . . . . .	15
MLM . . . . .	16
MLM_bic . . . . .	17
MLM_cv . . . . .	18
orderedLines . . . . .	18
plot_CTLE . . . . .	19
plot_mixtureReg . . . . .	19

*biscalew* 3

plot_mixtureRegList . . . . .	20
Rec_Lm . . . . .	20
rmr . . . . .	21
RobMixReg-class . . . . .	22
simuData . . . . .	22
simu_data_sparse . . . . .	23
simu_func . . . . .	23
simu_low . . . . .	24
TLE . . . . .	24

**Index** 26

---

*biscalew*                      *biscalew : Robust M-estimates for scale.*

---

**Description**

Tukey's bisquare family of functions.

**Usage**

`biscalew(t)`

**Arguments**

`t`                      Numerical input, usually residuals.

**Value**

bisquare weight for scale.

---

*bisquare*                      *bisquare : Robust estimates for mean.*

---

**Description**

Tukey's bisquare family of functions.

**Usage**

`bisquare(t, k = 4.685)`

**Arguments**

`t`                      Numerical input, usually residuals.  
`k`                      A constant tuning parameter, default is 4.685.

**Value**

A bi-square weight for mean.

---

blockMap	<i>Plot the coefficient matrix.</i>
----------	-------------------------------------

---

**Description**

Plot the coefficient matrix.

**Usage**

```
blockMap(rrr)
```

**Arguments**

rrr	The result from CSMR function
-----	-------------------------------

---

CCLE_data	<i>RobMixReg package built-in CCLE data.</i>
-----------	--

---

**Description**

The list which contain all the information to generate variables used in the real application.

**Usage**

```
CCLE_data
```

**Format**

A list whose length is 2:

**X** Gene expression dataset.

**Y** AUCC score.

---

colon_data	<i>RobMixReg package built-in Colon cancer data.</i>
------------	--

---

**Description**

The list which contain all the information to generate variables used in the real application.

**Usage**

```
colon_data
```

**Format**

A list whose length is 3:

**rnames** A string contains the name of binding protein and epigenetic regulator.

**x3** The gene expression profile of CREB3L1.

**y3** The methylation profile of cg16012690 on 299 colon adenocarcinoma patients.

**x2** x2

**y2** y2

**x1** x1

**y1** y1

---

compPlot	<i>The plot wrapper function.</i>
----------	-----------------------------------

---

**Description**

The plot wrapper function.

**Usage**

```
compPlot(type = "rlr", x, y, nc, inds_in, res)
```

**Arguments**

**type** The character to choose which type of plot to generate.

**x** The independent variables

**y** The external variable

**nc** The number of components

**inds\_in** A vector indicate the outlier samples.

**res** The result object returned by MLM function.

---

Compute\_Rbase\_SVD      *Compute the row space using SVD.*

---

**Description**

Compute the row space using SVD.

**Usage**

```
Compute_Rbase_SVD(bulk_data, tg_R1_lists_selected)
```

**Arguments**

bulk\_data      The bulk data..  
 tg\_R1\_lists\_selected      A list of the marker genes for several cell types.

**Value**

A matrix which each row span the row space using cell type specific marker genes.

---

CSMR      *The main function of the RBSL algorithm.*

---

**Description**

The main function of the RBSL algorithm.

**Usage**

```
CSMR(x, y, nit, nc, max_iter)
```

**Arguments**

x      The matrix  
 y      The external supervised variable.  
 nit      xxx?  
 nc      The component number in the mixture model.  
 max\_iter      The maximum iteration number.

**Value**

A list object consist of coefficient, clustering membership, data x, external variable y, predicted y based on regression model.

---

CSMR_one	<i>Perform the RBSL algorithm one times.</i>
----------	--

---

**Description**

Perform the RBSL algorithm one times.

**Usage**

```
CSMR_one(x, y, nit = 1, nc, max_iter)
```

**Arguments**

x	The matrix
y	The external supervised variable.
nit	xxx?
nc	The component number in the mixture model.
max_iter	The maximum iteration number.

**Value**

A list object consist of coefficient, clustering membership, data x, external variable y, predicted y based on regression model.

---

CSMR_predict	<i>The predict function of the CSMR algorithm.</i>
--------------	--

---

**Description**

The predict function of the CSMR algorithm.

**Usage**

```
CSMR_predict(CSMR_coffs, CSMR.model, xnew, ynew, singleMode = F)
```

**Arguments**

CSMR_coffs	The coefficient matrix.
CSMR.model	The trained model.
xnew	x variable.
ynew	y variable.
singleMode	A parameter to set the component to one.

**Value**

A list object consist of coefficient, clustering membership, data x, external variable y, predicted y based on regression model.

---

CSMR_train	<i>The train function of the CSMR algorithm.</i>
------------	--

---

**Description**

The train function of the CSMR algorithm.

**Usage**

```
CSMR_train(x, y, nit, nc, max_iter)
```

**Arguments**

x	The matrix
y	The external supervised variable.
nit	xxx
nc	The component number in the mixture model.
max_iter	The maximum iteration number.

**Value**

A list object consist of coefficient, clustering membership, data x, external variable y, predicted y based on regression model.

---

CTLERob	<i>CTLERob: Robust mixture regression based on component-wise adaptive trimming likelihood estimation.</i>
---------	--

---

**Description**

CTLERob performs robust linear regression with high breakdown point and high efficiency in each mixing components and adaptively remove the outlier samples.

**Usage**

```
CTLERob(formula, data, nit = 20, nc = 2, rlr_method = "ltsReg")
```

```
## S4 method for signature 'formula,ANY,ANY,numeric'
CTLERob(formula, data, nit = 20,
         nc = 2, rlr_method = "ltsReg")
```



**Arguments**

formula	A symbolic description of the model to be fit.
data	A data frame containing the predictor and response variables, where the last column is the response variable.
nit	Number of iterations.
nc	Number of mixture components.
r1r_method	The regression methods, default is 'ItsReg'.

---

denLp                      *denLp : Density function for Laplace distribution.*

---

**Description**

Laplace distribution.

**Usage**

```
denLp(rr, sig)
```

**Arguments**

rr	Shift from the location parameter
sig	Scale parameter.

**Value**

Laplace density.

---

DeOut                      *DeOut : Detect outlier observations.*

---

**Description**

Detect outlier observations from a vector.

**Usage**

```
DeOut(daData, method)
```

**Arguments**

daData	A numerical vector.
method	Choose from '3sigma', 'hampel' and 'boxplot'.

**Value**

indices of outlier observations.

---

flexmix_2	<i>flexmix_2: Multiple runs of MLE based mixture regression to stabilize the output.</i>
-----------	--

---

**Description**

Mixture regression based on MLE could be unstable when assuming unequal variance. Multiple runs of flexmix is performed to stabilize the results.

**Usage**

```
flexmix_2(formula, data1, k, mprior)
```

**Arguments**

formula	A symbolic description of the model to be fit.
data1	A data frame containing the predictor and response variables, where the last column is the response variable.
k	Number of mixture components.
mprior	A numeric number in (0,1) that specifies the minimum proportion of samples in each mixing components.

**Value**

A S4 object of flexmix class. xxx

---

gaussData	<i>RobMixReg package built-in gaussian example data.</i>
-----------	--

---

**Description**

A dataset generated from gaussian distribution in RobMixReg package.

**Usage**

```
gaussData
```

**Format**

A data frame with 100 rows and 3 variables:

**x** x variable

**y** y variable

**c** cluster information

---

lars.lsa	<i>lars variant for LSA.</i>
----------	------------------------------

---

**Description**

lars variant for LSA.

**Usage**

```
lars.lsa(Sigma0, b0, intercept, n, type = c("lasso", "lar"),
        eps = .Machine$double.eps, max.steps)
```

**Arguments**

Sigma0	The parameter.
b0	The intercept of the regression line.
intercept	The bool variable of whether consider the intercept situation
n	The number of data point.
type	Regression options, choose form "lasso" or "lar".
eps	The converge threshold defined by the machine.
max.steps	The maximum iteration times to stop.

**Value**

object.

**Author(s)**

Reference Wang, H. and Leng, C. (2006) and Efron et al. (2004).

---

logLik_mixtureReg	<i>Obtain Log-likelihood from a mixtureReg Object</i>
-------------------	---

---

**Description**

S3 method for class 'mixtureReg'. However, it doesn't return a 'logLik' object. For simlicity, it returns a 'numeric' value.

**Usage**

```
logLik_mixtureReg(mixtureModel)
```

**Arguments**

mixtureModel	mixtureReg object, typically result from 'mixtureReg()'.
--------------	--

**Value**

Return a numeric value of log likelihood.

---

lsa	<i>Least square approximation. This version Oct 19, 2006.</i>
-----	---

---

**Description**

Least square approximation. This version Oct 19, 2006.

**Usage**

```
lsa(obj)
```

**Arguments**

obj            lm/glm/coxph or other object.

**Value**

beta.ols: the MLE estimate ; beta.bic: the LSA-BIC estimate ; beta.aic: the LSA-AIC estimate.

**Author(s)**

Reference Wang, H. and Leng, C. (2006) and Efron et al. (2004).

---

mixlinrb_bi	<i>mixlinrb_bi: mixlinrb_bione estimates the mixture regression parameters robustly using bisquare function based on multiply initial value.</i>
-------------	--

---

**Description**

An EM-type of parameter estimation by replacing the least square estimation in the M-step with a robust criterion.

**Usage**

```
mixlinrb_bi(formula, data, nc = 2, nit = 200)

## S4 method for signature 'formula,ANY,numeric,numeric'
mixlinrb_bi(formula, data,
             nc = 2, nit = 20)
```

**Arguments**

formula	A symbolic description of the model to be fit.
data	A data frame containing the predictor and response variables, where the last column is the response variable.
nc	Number of mixture components.
nit	Number of iterations for biSauqre method.

**Value**

Estimated coefficients of all components.

---

mixlinrb_bione	<i>mixlinrb_bione : mixlinrb_bione estimates the mixture regression parameters robustly using bisquare function based on one initial value.</i>
----------------	---

---

**Description**

An EM-type of parameter estimation by replacing the least square estimation in the M-step with a robust criterion.

**Usage**

```
mixlinrb_bione(formula, data, nc = 2)
```

**Arguments**

formula	A symbolic description of the model to be fit.
data	A data frame containing the predictor and response variables, where the last column is the response variable.
nc	Number of mixture components.

**Value**

Estimated coefficients of all components.

---

mixLp	<i>mixLp : mixLp_one estimates the mixture regression parameters robustly using Laplace distribution based on multiply initial value..</i>
-------	--

---

### Description

mixLp estimates the mixture regression parameters robustly using bisquare function based on multiple initial values. The solution is found by the modal solution.

### Usage

```
mixLp(formula, data, nc=2, nit=200)

## S4 method for signature 'formula,ANY,numeric,numeric'
mixLp(formula, data, nc = 2,
      nit = 20)
```

### Arguments

formula	A symbolic description of the model to be fit.
data	A data frame containing the predictor and response variables, where the last column is the response variable.
nc	Number of mixture components.
nit	Number of iterations

### Value

Estimated coefficients of all components.

### Examples

```
library("RobMixReg")
formula01=as.formula("y~x")
x=(gaussData$x);y=as.numeric(gaussData$y);
example_data01=data.frame(x,y)
res = mixLp(formula01, example_data01, nc=2, nit=20)
```

---

mixLp_one	<i>mixLp_one : mixLp_one estimates the mixture regression parameters robustly using Laplace distribution based on one initial value.</i>
-----------	--

---

**Description**

Robust mixture regression assuming that the error terms follow a Laplace distribution.

**Usage**

```
mixLp_one(formula, data, nc = 2)
```

**Arguments**

formula	A symbolic description of the model to be fit.
data	A data frame containing the predictor and response variables, where the last column is the response variable.
nc	Number of mixture components.

**Value**

Estimated coefficients of all components.

---

mixtureReg	<i>Function to Fit Mixture of Regressions</i>
------------	---

---

**Description**

The main function in this package.

**Usage**

```
mixtureReg(regData, formulaList, xName = NULL, yName = NULL,
  mixingProb = c("Constant", "loess"), initialWList = NULL,
  epsilon = 1e-08, max_iter = 10000, max_restart = 15,
  min_lambda = 0.01, min_sigmaRatio = 0.1, silently = TRUE)
```

**Arguments**

regData	data frame used in fitting model.
formulaList	a list of the regression components that need to be estimated.
xName	character; Name used to pick x variable from data.
yName	character; Name used to pick y variable from data.

mixingProb	character; Specify how the mixing probabilities are estimated in the M step. "Constant" specifies a constant mixing probabilities; "loess" specifies predictor dependent mixing probabilities obtained by loess smoothing.
initialWList	a list of weights guesses (provided by user). Typically this is not used, unless the user has a good initial guess.
epsilon	a small value that the function consider as zero. The value is used in determine matrix singularity and in determine convergence.
max_iter	the maximum number of iterations.
max_restart	the maximum number of restart before giving up.
min_lambda	a value used to ensure estimated mixing probabilities (lambda's) are not too close to zero.
min_sigmaRatio	a value used to prevent estimated variaces of any regression component from collapsing to zero.
silently	a switch to turn off the screen printout.

**Value**

A class 'mixtureReg' object.

**Author(s)**

The mixtureReg package is developed by Tianxia Zhou on github. All right reserved by Tianxia Zhou.

---

 MLM

*The main function of mining the latent relationship among variables.*

---

**Description**

The main function of mining the latent relationship among variables.

**Usage**

```
MLM(ml.method = "rlr", rmr.method = "cat",
    b.formulaList = list(formula(y ~ x), formula(y ~ 1)), formula = y ~
    x, nit = 1, nc = 2, x = NULL, y = NULL, max_iter = 50,
    tRatio = 0.05)
```

**Arguments**

ml.method	The option to select the four methods in vignette.
rmr.method	The option to select the robust mixture regression method.
b.formulaList	The case b require the user provide the formula list. This enable the flexible mixture regression.



formula	The linear relationship between two variables.
nit	Number of iterations for CTLE, mixbi, mixLp.
nc	Number of mixture components.
x	The matrix x of the high dimension situation.
y	The external outcome variable.
max_iter	Maximum iteration for TLE method.
tRatio	The ratio of the outliers in the TLE robust mixture regression method.

**Value**

Main result object.

---

MLM_bic	<i>Model selection function for low dimension data.</i>
---------	---

---

**Description**

Model selection function for low dimension data.

**Usage**

```
MLM_bic(ml.method = "rlr", x, y, nc = 1, formulaList = NULL, K = 2)
```

**Arguments**

ml.method	The parameter to choose the fitted model for calculating the BIC
x	x variable.
y	y variable.
nc	The component number for low dimensional feature
formulaList	The list of target formular
K	The component number for high dimensional feature

**Value**

BIC value.

---

MLM_cv	<i>Cross validation (fold-5) function for high dimension data.</i>
--------	--

---

**Description**

Cross validation (fold-5) function for high dimension data.

**Usage**

```
MLM_cv(x = NULL, y = NULL, nit = 1, nc = 2, max_iter = 50)
```

**Arguments**

x	x variable.
y	y variable.
nit	Iteration number.
nc	The number of component.
max_iter	Maximum iteration.

**Value**

The correlation between y and y\_hat based on five fold cross validation.

---

orderedLines	<i>Sort by X Coordinates and Add Line to a Plot</i>
--------------	---

---

**Description**

Rearrange X and Y coordinates before calling "lines()" function.

**Usage**

```
orderedLines(x, y, ...)
```

**Arguments**

x	X coordinate vectors of points to join.
y	Y coordinate vectors of points to join.
...	Further graphical parameters.

---

plot_CTLE	<i>plot_CTLE: Plot the mixture/single regression line(s) in a simply function.</i>
-----------	--

---

**Description**

CTLERob performs robust linear regression with high breakdown point and high efficiency in each mixing components and adaptively remove the outlier samples.

**Usage**

```
plot_CTLE(formula, data, nc = 2, inds_in)

## S4 method for signature 'formula,ANY,numeric'
plot_CTLE(formula, data, nc = 2, inds_in)
```

**Arguments**

formula	A symbolic description of the model to be fit.
data	A data frame containing the predictor and response variables, where the last column is the response variable.
nc	Number of mixture components.
inds_in	The index of the point which belongs to the current regression line.

---

plot_mixtureReg	<i>Plot Fit and Mixing Probability of a mixtureReg Object</i>
-----------------	---

---

**Description**

S3 plot method for class 'mixtureReg'.

**Usage**

```
plot_mixtureReg(mixtureModel, which = 1:2, xName = NULL,
  yName = NULL, xlab = NULL, ylab = NULL, ...)
```

**Arguments**

mixtureModel	mixtureReg object, typically result from 'mixtureReg()'.
which	numeric; choose which plot to display. '1' gives a plot of fit; '2' gives a plot of mixing probability.
xName	character; Name used to pick x variable from data.
yName	character; Name used to pick y variable from data.
xlab	character; label that should be put on the x axis.
ylab	character; label that should be put on the y axis.
...	Further graphical parameters.

---

plot\_mixtureRegList     *Plot a List of mixtureReg Objects*

---

**Description**

Feed in a list of mixtureReg models and get an overlaid plot.

**Usage**

```
plot_mixtureRegList(mixtureRegList, xName = NULL, yName = NULL, ...)
```

**Arguments**

mixtureRegList   a list of multiple mixtureReg objects.  
xName             character; Name used to pick x variable from data.  
yName             character; Name used to pick y variable from data.  
...                Further graphical parameters.

---

Rec\_Lm             *Adaptive lasso.*

---

**Description**

Adaptive lasso.

**Usage**

```
Rec_Lm(XX, yy)
```

**Arguments**

XX                The independent variable.  
yy                The dependent variable.

**Value**

A list object consist of index of selected variable and coefficient for all variables.

---

 rnr

---

*The main function of Robust Mixture Regression using five methods.*


---

### Description

The main function of Robust Mixture Regression using five methods.

### Usage

```
rnr(lr.method = "flexmix", formula = NULL, data = NULL, nc = 2,
    nit = 20, tRatio = 0.05, MaxIt = 200)
```

### Arguments

<code>lr.method</code>	A robust mixture regression method to be used. Should be one of "flexmix", "TLE", "CTLERob", "mixbi", "mixLp".
<code>formula</code>	A symbolic description of the model to be fit.
<code>data</code>	A data frame containing the predictor and response variables, where the last column is the response variable.
<code>nc</code>	Number of mixture components.
<code>nit</code>	Number of iterations for CTLE, mixbi, mixLp.
<code>tRatio</code>	Trimming proportion for TLE method.
<code>MaxIt</code>	Maximum iteration for TLE method.

### Value

An S4 object about the regression result.

### Examples

```
library(RobMixReg)
#library(robust)
library(flexmix)
library(robustbase)
library(MASS)
library(gtools)
# gaussData
x=(gaussData$x);y=as.numeric(gaussData$y);
formula01=as.formula("y~x")
example_data01=data.frame(x,y)
res_rnr = rnr(lr.method='flexmix', formula=formula01, data=example_data01)
res_rnr = rnr(lr.method='CTLERob', formula=formula01, data=example_data01)
```

---

RobMixReg-class	<i>Class RobMixReg.</i>
-----------------	-------------------------

---

**Description**

Class RobMixReg defines a robust mixture regression class as a S4 object.

**Slots**

`inds_in` The indices of observations used in the parameter estimation.

`indout` The indices of outlier samples, not used in the parameter estimation.

`ctleclusters` The cluster membership of each observation.

`compcoef` Regression coefficients for each component.

`compvals` Component p values.

`compwww` The posterior of the clustering.

`call` Call function.

---

simuData	<i>RobMixReg package built-in simulated example data.</i>
----------	---

---

**Description**

A simulation dataset from RobMixReg package. This simulation dataset is in dimension 2 and ground truth (include outliers label) of the cluster information also generated.

**Usage**

```
simuData
```

**Format**

A data frame with 500 rows and 5 variables:

**X1** X1 variable

**X2** X2 variable

**y** y variable

**c** cluster information

**outlier** outlier indicator

---

simu_data_sparse	<i>Simulate high dimension data for RBSL algorithm validation.</i>
------------------	--

---

**Description**

Simulate high dimension data for RBSL algorithm validation.

**Usage**

```
simu_data_sparse(n, bet, pr, sigma)
```

**Arguments**

n	Patient number.
bet	The coefficient matrix.
pr	A vector of probability threshold which simulate the sampling based on uniform distribution.
sigma	A vector of noise level. The length should be equal to the component number.

**Value**

A list object consist of x, y, true cluster label.

---

simu_func	<i>The simulation function for low/high dimensional space.</i>
-----------	--

---

**Description**

The simulation function for low/high dimensional space.

**Usage**

```
simu_func(beta, sigma, alpha = NULL, n = 400)
```

**Arguments**

beta	The slope vector for low dimensional space or matrix for high dimensional space.
sigma	A vector whose k-th element is the standard deviation for the k-th regression component.
alpha	The parameter to control the number of outliers for low dimensional space.
n	The sample number for high dimensional data.

**Value**

A list object.

---

simu_low	<i>The simulation function for low dimensional space.</i>
----------	---

---

**Description**

The simulation function for low dimensional space.

**Usage**

```
simu_low(beta, inter, alpha = NULL)
```

**Arguments**

beta	The slope vector.
inter	The intercept vector.
alpha	The parameter to control the number of outliers.

**Value**

A list object consists of the x variable in low dimensional space and the external y variable.

---

TLE	<i>TLE: robust mixture regression based on trimmed likelihood estimation.</i>
-----	---

---

**Description**

The algorithm fits a mixture regression model after trimming a proportion of the observations, given by tRatio.

**Usage**

```
TLE(formula, data, nc = 2, tRatio, MaxIt = 200)

## S4 method for signature 'formula,ANY,numeric,numeric,numeric'
TLE(formula, data,
     nc = 2, tRatio, MaxIt = 200)
```

**Arguments**

formula	A symbolic description of the model to be fit.
data	A data frame containing the predictor and response variables, where the last column is the response variable.
nc	Number of mixture components.
tRatio	Trimming proportion.
MaxIt	Maximum iteration.



**Value**

A S4 object of RobMixReg class.

**Examples**

```
library("RobMixReg")
formula01=as.formula("y~x")
x=(gaussData$x);y=as.numeric(gaussData$y);
example_data01=data.frame(x,y)

res = TLE(formula01,example_data01, nc=2,tRatio=0.05,MaxIt=200)
```

# Index

- \* **datasets**
  - CCLC\_data, [4](#)
  - colon\_data, [5](#)
  - gaussData, [10](#)
  - simuData, [22](#)
- biscalew, [3](#)
- bisquare, [3](#)
- blockMap, [4](#)
- CCLC\_data, [4](#)
- colon\_data, [5](#)
- compPlot, [5](#)
- Compute\_Rbase\_SVD, [6](#)
- CSMR, [6](#)
- CSMR\_one, [7](#)
- CSMR\_predict, [7](#)
- CSMR\_train, [8](#)
- CTLE, formula, ANY, numeric, numeric, numeric-method (TLE), [24](#)
- CTLE, formula, ANY, numeric, numeric-method (mixLp), [14](#)
- CTLERob, [8](#)
- CTLERob, formula, ANY, ANY, numeric, ANY-method (CTLERob), [8](#)
- CTLERob, formula, ANY, ANY, numeric-method (CTLERob), [8](#)
- CTLERob, formula, ANY, numeric, ANY (plot\_CTLE), [19](#)
- denLp, [9](#)
- DeOut, [9](#)
- flexmix\_2, [10](#)
- gaussData, [10](#)
- lars.lsa, [11](#)
- logLik\_mixtureReg, [11](#)
- lsa, [12](#)
- mixlinrb\_bi, [12](#)
- mixlinrb\_bi, formula, ANY, numeric, numeric-method (mixlinrb\_bi), [12](#)
- mixlinrb\_bione, [13](#)
- mixLp, [14](#)
- mixLp, formula, ANY, numeric, numeric-method (mixLp), [14](#)
- mixLp\_one, [15](#)
- mixtureReg, [15](#)
- MLM, [16](#)
- MLM\_bic, [17](#)
- MLM\_cv, [18](#)
- orderedLines, [18](#)
- plot\_CTLE, [19](#)
- plot\_CTLE, formula, ANY, numeric-method (plot\_CTLE), [19](#)
- plot\_mixtureReg, [19](#)
- plot\_mixtureRegList, [20](#)
- Rec\_Lm, [20](#)
- rnr, [21](#)
- RobMixReg-class, [22](#)
- simu\_data\_sparse, [23](#)
- simu\_func, [23](#)
- simu\_low, [24](#)
- simuData, [22](#)
- TLE, [24](#)
- TLE, formula, ANY, numeric, numeric, numeric-method (TLE), [24](#)