

# Package ‘SeleMix’

July 2, 2014

**Type** Package

**Title** Selective Editing via Mixture models

**Version** 0.9.1

**Date** 2013-12-10

**Author** Ugo Guarnera, M. Teresa Buglielli

**Maintainer** Teresa Buglielli <bugliell@istat.it>

**Description** Detection of outliers and influential errors using a latent variable model.

**Depends** mvtnorm, Ecdat, xtable

**License** EUPL

**LazyLoad** yes

**LazyData** yes

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2013-12-12 14:06:08

## R topics documented:

ex1.data . . . . .	2
ex2.data . . . . .	2
ml.est . . . . .	3
pred.y . . . . .	5
sel.edit . . . . .	7
sel.pairs . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

`ex1.data`*Example data frame*

---

**Description**

Simulated data from a Gaussian contamination model

**Usage**

```
data(ex1.data)
```

**Format**

A data frame with 500 observations and 2 variables (X1,Y1).

X1 error-free variable (numeric)

Y1 contaminated variable (numeric)

**Details**

Data have been generated by a Gaussian model. The variable Y1 has been contaminated with parameters  $B=(-0.26, 1.26)$ ,  $\sigma=1.21$ ,  $w=0.05$ ,  $\lambda=10$ .

**Examples**

```
data(ex1.data)
```

---

`ex2.data`*Example Data for package SeleMix*

---

**Description**

Simulated data from a Gaussian contaminated model

**Usage**

```
data(ex2.data)
```

**Format**

A data frame with 500 observations on the following 2 variables.

Y1 first numeric contaminated variable

Y2 second numeric contaminated variable

## Details

Data have been simulated by Gaussian contamination model with two contaminated variables (Y1,Y2) with parameters  $B=(1.03, 0.96)$ ,  $\sigma=\text{matrix}(c(1.22, 1.42,1.42, 2.89),2,2)$ ,  $w=0.05$ ,  $\lambda=10$ .

## Examples

```
data(ex2.data)
```

---

```
ml.est
```

```
Fitting Contamination Model
```

---

## Description

Provides ML estimates of a Gaussian contamination model.

## Usage

```
ml.est (y, x=NULL, model = "LN", lambda=3, w=0.05,
        lambda.fix=FALSE, w.fix=FALSE, eps=1e-7,
        max.iter=500, t.outl=0.5, graph=FALSE)
```

## Arguments

y	matrix or data frame containing the response variables
x	optional matrix or data frame containing the error free covariates
model	data distribution: LN = lognormal(default), N=normal
lambda	starting value for the variance inflation factor (default=3)
w	starting value for the proportion of contaminated data (default=0.05)
lambda.fix	logical. TRUE if lambda is known
w.fix	logical. TRUE if w is known
eps	epsilon : tolerance parameter for the log-likelihood convergence (default=1e-7)
max.iter	maximum number of EM iterations (default=500)
t.outl	threshold value for posterior probabilities of identifying outliers (default=0.5)
graph	logical. TRUE to display graphics (default=FALSE)

## Details

This function provides the parameter estimates of a contamination model where a set of  $y$  variables is assumed to depend on a (possibly empty) set of covariates ( $x$  variables) through a mixture of two linear regressions with Gaussian residuals. The covariance matrices of the two mixture components are assumed to be proportional (the proportionality constant being  $\lambda$ ). In case of no  $x$  variables a mixture of two Gaussian distribution is estimated. BIC and AIC scores (`bic.aic`) are returned corresponding to both standard Gaussian model and contamination model in order to help the user to avoid possible over-parametrisation.

According to the estimated model parameters, a matrix of predictions of ‘true’  $y$  values (`ypred`) is computed. To each unit in the dataset, a flag (`outlier`) is assigned taking value 0 or 1 depending on whether the posterior probability of being erroneous ( $\tau$ ) is greater than the user specified threshold (`t.out1`).

The model is estimated using complete observations. Missing values in the  $x$  variables are not allowed. However,  $y$  variables can be partly observed. Robust predictions of  $y$  variables are provided even when they are not observed. A vector of missing pattern (`pattern`) indicates which item is observed and which is missing.

In case the option ‘`model = LN`’ is specified, each zero value is changed in  $1e-7$  and a warning is returned.

In order to graphically monitor EM algorithm, a scatter plot is showed where outliers are depicted as long as they are identified. The trajectory of the  $\lambda$  parameter is also showed until convergence.

## Value

`ml.est` returns a list containing the following components:

<code>ypred</code>	matrix of predicted values for $y$ variables
<code>B</code>	matrix of estimated regression coefficients
<code>sigma</code>	estimated covariance matrix
<code>lambda</code>	estimated variance inflation factor
<code>w</code>	estimated proportion of erroneous data
<code>tau</code>	vector of posterior probabilities of being contaminated
<code>outlier</code>	1 if the observation is classified as an outlier, 0 otherwise
<code>pattern</code>	vector of non-response patterns for $y$ variables: 0 = missing, 1 = present value
<code>is.conv</code>	logical value: TRUE if the EM algorithm has converged
<code>n.iter</code>	number of iterations of EM algorithm
<code>bic.aic</code>	Bayesian Information Criterion and Akaike Information Criterion for contaminated and non contaminated Gaussian models

## Author(s)

M. Teresa Buglielli <bugliell@istat.it>, Ugo Guarnera <guarnera@istat.it>

## References

Di Zio, M., Guarnera, U. (2013) "A Contamination Model for Selective Editing", Journal of Official Statistics. Volume 29, Issue 4, Pages 539-555 (<http://dx.doi.org/10.2478/jos-2013-0039>).

Buglielli, M.T., Di Zio, M., Guarnera, U. (2010) "Use of Contamination Models for Selective Editing", European Conference on Quality in Survey Statistics Q2010, Helsinki, 4-6 May 2010

## Examples

```
# Parameter estimation with one contaminated variable and one covariate
data(ex1.data)
ml.par <- ml.est(y=ex1.data[, "Y1"], x=ex1.data[, "X1"], graph=TRUE)
str(ml.par)
sum(ml.par$outlier) # number of outliers
# Parameter estimation with two contaminated variables and no covariates
data(ex2.data)
par.joint <- ml.est(y=ex2.data, x=NULL, graph=TRUE)
sum(par.joint$outlier) # number of outliers
```

---

pred.y

*Prediction of y variables*

---

## Description

Provides predictions of y variables according to a Gaussian contamination model

## Usage

```
pred.y (y, x=NULL, B, sigma, lambda, w, model="LN", t.outl=0.5)
```

## Arguments

y	matrix or data frame containing the response variables
x	optional matrix or data frame containing the error free covariates
B	matrix of regression coefficients
sigma	covariance matrix
lambda	variance inflation factor
w	proportion of erroneous data
model	data distribution: LN = lognormal(default), N=normal
t.outl	threshold value for posterior probabilities of identifying outliers (default=0.5)

## Details

This function provides expected values of a set of variables ( $y1.p, y2.p, \dots$ ) according to a mixture of two regression models with Gaussian residuals (see [ml.est](#)). If no covariates are available (x variables), a two component Gaussian mixture is used. Expected values (predictions) are computed on the base of a set of parameters of appropriate dimensions ( $B, \sigma, \lambda, w$ ) and (possibly) a matrix (or data frame) containing the error-free x variables.

Missing values in the x variables are not allowed. However, robust predictions of y variables are also provided when these variables are not observed. A vector of missing pattern (`pattern`) indicates which item is observed and which is missing.

For each unit in the data set the posterior probability of being erroneous ( $\tau$ ) is computed and a flag (`outlier`) is provided taking value 0 or 1 depending on whether  $\tau$  is greater than the user specified threshold (`t.out1`).

## Value

`pred.y` returns a data frame containing the following columns:

<code>y1.p, y2.p, ...</code>	predicted values for y variables
<code>tau</code>	posterior probabilities of being contaminated
<code>outlier</code>	1 if the observation is classified as an outlier, 0 otherwise
<code>pattern</code>	non-response patterns for y variables: 0 = missing, 1 = present value

## Author(s)

M. Teresa Buglielli <bugliell@istat.it>, Ugo Guarnera <guarnera@istat.it>

## References

Buglielli, M.T., Di Zio, M., Guarnera, U. (2010) *"Use of Contamination Models for Selective Editing"*, European Conference on Quality in Survey Statistics Q2010, Helsinki, 4-6 May 2010

## Examples

```
# Parameter estimation with one contaminated variable and one covariate
data(ex1.data)
# Parameters estimated applying ml.est to \code{ex1.data}
B1 <- as.matrix(c(-0.152, 1.215))
sigma1 <- as.matrix(1.25)
lambda1 <- 15.5
w1 <- 0.0479

# Variable prediction
ypred <- pred.y (y=ex1.data[, "Y1"], x=ex1.data[, "X1"], B=B1,
               sigma=sigma1, lambda=lambda1, w=w1, model="LN", t.out1=0.5)
# Plot ypred vs Y1
sel.pairs(cbind(ypred[, 1, drop=FALSE], ex1.data[, "Y1", drop=FALSE]),
```

```
outl=ypred[,"outlier"])
```

---

 sel.edit

*Influential Error Detection*


---

## Description

Computes the score function and identifies influential errors

## Usage

```
sel.edit (y, ypred, wgt=rep(1,nrow(as.matrix(y))),
         tot=colSums(ypred * wgt), t.sel=0.01)
```

## Arguments

y	matrix or data frame containing the response variables
ypred	matrix of predicted values for y variables
wgt	optional vector of sampling weights (default=1)
tot	optional vector containing reference estimates of totals for the y variables. If omitted, it is computed as the (possibly weighted) sum of predicted values
t.sel	optional vector of threshold values, one for each variable, for selective editing (default=0.01)

## Details

This function ranks observations (*rank*) according to the importance of their potential errors. The order is made with respect to the global score function values (*global.score*). The function also selects the units to be edited (*sel*) so that the expected residual error of all variables is below a prefixed level of accuracy (*t.sel*). The global score (*global.score*) is the maximum of the local scores computed for each variable (*y1.score*, *y2.score*, ...). The local scores are defined as a weighted (*weights*) absolute difference between the observed (*y1*, *y2*, ...) and the predicted values (*y1.p*, *y2.p*, ...) standardised with respect to the reference total estimates (*tot*).

The selection of the units to be edited because affected by an influential error (*sel=1*) is made according to a two-step algorithm:

- 1) order the observations with respect to the *global.score* (decreasing order);
- 2) select the first *k* units such that, from the (*k+1*)th to the last observation, all the residual errors (*y1.reserr*, *y2.reserr*, ...) for each variable are below *t.sel*.

The function provides also an indicator function (*y1.sel*, *y2.sel*, ...) reporting which variables contain an influential errors in a unit selected for the revision.

**Value**

sel.edit returns a data matrix containing the following columns:

y1, y2, ...	observed variables
y1.p, y2.p, ...	predictions of y variables
weights	sampling weights
y1.score, y2.score, ...	local scores
global.score	global score
y1.reserr, y2.reserr, ...	residual errors
y1.sel, y2.sel, ...	influential error flags
rank	rank according to global score
sel	1 if the observation contains an influential error, 0 otherwise

**Author(s)**

M. Teresa Buglielli <bugliell@istat.it>, Ugo Guarnera <guarnera@istat.it>

**References**

Di Zio, M., Guarnera, U. (2013) "A Contamination Model for Selective Editing", Journal of Official Statistics. Volume 29, Issue 4, Pages 539-555 (<http://dx.doi.org/10.2478/jos-2013-0039>).

Buglielli, M.T., Di Zio, M., Guarnera, U. (2010) "Use of Contamination Models for Selective Editing", European Conference on Quality in Survey Statistics Q2010, Helsinki, 4-6 May 2010.

**Examples**

```
# Example 1
# Parameter estimation with one contaminated variable and one covariate
data(ex1.data)
ml.par <- ml.est(y=ex1.data[, "Y1"], x=ex1.data[, "X1"])
# Detection of influential errors
sel <- sel.edit(y=ex1.data[, "Y1"], ypred=ml.par$ypred)
head(sel)
sum(sel[, "sel"])
# orders results for decreasing importance of score
sel.ord <- sel[order(sel[, "rank"]), ]
# adds columns to data
ex1.data <- cbind(ex1.data, tau=ml.par$tau, outlier=ml.par$outlier,
                 sel[, c("rank", "sel")])
# plot of data with outliers and influential errors
sel.pairs(ex1.data[, c("X1", "Y1")], outl=ml.par$outlier, sel=sel[, "sel"])
# Example 2
data(ex2.data)
```



```
m1.par <- ml.est(y=ex2.data)
sel <- sel.edit(y=ex2.data, ypred=m1.par$ypred)
sel.pairs(ex2.data,outl=m1.par$outlier, sel=sel[, "sel"])
```

---

sel.pairs

*Scatterplot Matrix*

---

### Description

A scatterplot matrix with outlier and influential errors is produced.

### Usage

```
sel.pairs(x, outl = rep(0, nrow(x)), sel = rep(0, nrow(x)),
          labs = NULL, log = TRUE, legend=TRUE)
```

### Arguments

x	matrix or data frame of the coordinates of points
outl	vector identifying outliers (1 or TRUE means outlier)
sel	vector identifying influential errors (1 or TRUE means influential error)
labs	names of the variables
log	if TRUE logarithm of x are plotted
legend	if TRUE a legend is added to first boxplot

### Details

The  $ij$ th scatterplot contains  $x[,i]$  plotted against  $x[,j]$ . Outliers are represented as blue circles, influential errors as red circles and points that are both outlier and influential error as cyan circles.

# Index

## \*Topic **datasets**

ex1.data, [2](#)

ex2.data, [2](#)

ex1.data, [2](#)

ex2.data, [2](#)

m1.est, [3](#), [6](#)

pred.y, [5](#)

sel.edit, [7](#)

sel.pairs, [9](#)