Package ‘HeckmanEM’

October 12, 2022

Type Package

Title Fit Normal or Student-t Heckman Selection Models

Version 0.1.3

Author Marcos Prates [aut, cre, trl],
Victor Lachos [aut],
Dipak Dey [aut]

Maintainer Marcos Prates `<marcosop@est.ufmg.br>`

Description Maximum likelihood estimation by an EM algorithm of Heckman-type sample selection Normal or Student-t models. The reference is Lachos, Prates and Dey (2020) <doi:10.1016/j.jmva.2021.104737>.

Imports mvtnorm (>= 1.1-0), sampleSelection (>= 1.2-6), MomTrunc (>= 5.79), PerformanceAnalytics (>= 2.0.4)

License GPL (>= 2.0)

Encoding UTF-8

RoxygenNote 7.2.1

NeedsCompilation no

Repository CRAN

Date/Publication 2022-09-12 09:22:56 UTC

R topics documented:

`HeckmanEM` *Fit the Normal or Student-t Heckman Selectio model.*

Description

‘HeckmanEM()‘ fit the Heckman selection model.

Usage

```
HeckmanEM(
  y,
  x,
  w,
  cc,
  nu = 4,
  family = "T",
  error = 1e-05,
  iter.max = 500,
  im = TRUE,
  criteria = TRUE,
  verbose = TRUE
)
```

Arguments

`y`	A response vector.
`x`	A covariate matrix for the response y.
`w`	A covariate matrix for the missing indicator cc.
`cc`	A missing incidator vector (1=obserced, 0=missing) .
`nu`	The initial value for the degrees of freedom.
`family`	The family to be used (Normal or T).
`error`	The abslute convergence error for the EM stopping rule.
`iter.max`	The maximum number of iterations for the EM algorithm.
`im`	TRUE/FALSE, boolean to decide if the standard erros of the parameters should be computed.
`criteria`	TRUE/FALSE, boolean to decide if the model selection criteria should be computed.
`verbose`	TRUE/FALSE, boolean to decide if the progress should be printed in the screen.

Value

An object of the class HeckmanEM with all the outputs provided from the function.

Examples

```
n <- 100
family <- "T"
nu <- 4
rho <- .6
cens <- .25

set.seed(20200527)
w <- cbind(1,runif(n,-1,1),rnorm(n))
x <- cbind(w[,1:2])
c <- qt(cens, df=nu)

sigma2 <- 1

beta <- c(1,0.5)
gamma <- c(1,0.3,-.5)
gamma[1] <- -c*sqrt(sigma2)

set.seed(1)
datas <- rHeckman(x,w,beta,gamma,sigma2,rho,nu,family=family)
y <- datas$y
cc <- datas$cc

# Normal EM
res.N <- HeckmanEM(y, x, w, cc, nu = 4, family="Normal", error = 1e-05, iter.max = 50,
                   im=TRUE, criteria = TRUE)
# Student-t: EM
res.T <- HeckmanEM(y, x, w, cc, nu = 4, family="T", error = 1e-05, iter.max = 50,
                   im=TRUE, criteria = TRUE)
```

`HeckmanEM.criteria` *Model selection criteria for the Heckman Selectio model*

Description

‘HeckmanEM.criteria()‘ calculate the AIC, AICc, BIC selection criteria for the fitted Heckman selection model.

Usage

```
HeckmanEM.criteria(obj)
```

Arguments

`obj`	An object of the class HeckmanEM.

Value

The calculated AIC, AICc, and BIC for the parameters of the fitted model.

Examples

```
n <- 100
family <- "T"
nu <- 4
rho <- .6
cens <- .25

set.seed(20200527)
w <- cbind(1,runif(n,-1,1),rnorm(n))
x <- cbind(w[,1:2])
c <- qt(cens, df=nu)

sigma2 <- 1

beta <- c(1,0.5)
gamma <- c(1,0.3,-.5)
gamma[1] <- -c*sqrt(sigma2)

set.seed(1)
datas <- rHeckman(x,w,beta,gamma,sigma2,rho,nu,family=family)
y <- datas$y
cc <- datas$cc

res <- HeckmanEM(y, x, w, cc, nu = 4, family = "Normal", error = 1e-05, iter.max = 500,
                 im = TRUE, criteria = FALSE)
cr <- HeckmanEM.criteria(res)
```

`HeckmanEM.envelope` *Envelope for the Heckman Selectio model.*

Description

‘HeckmanEM.envelope()‘ plot the envelope for the fitted Heckman selection model.

Usage

```
HeckmanEM.envelope(obj, envelope = 0.95, ...)
```

Arguments

`obj`	An object of the class HeckmanEM.
`envelope`	The envelope corverage percertange.
`...`	Other option for chart.QQPlot from PerformanceAnalytics package.

Value

A residual plot of the fitted data and its envelope.

Examples

```
n <- 100
family <- "T"
nu <- 4
rho <- .6
cens <- .25

set.seed(20200527)
w <- cbind(1,runif(n,-1,1),rnorm(n))
x <- cbind(w[,1:2])
c <- qt(cens, df=nu)

sigma2 <- 1

beta <- c(1,0.5)
gamma <- c(1,0.3,-.5)
gamma[1] <- -c*sqrt(sigma2)

set.seed(1)
datas <- rHeckman(x,w,beta,gamma,sigma2,rho,nu,family=family)
y <- datas$y
cc <- datas$cc

res <- HeckmanEM(y, x, w, cc, nu = 4, family = "Normal", error = 1e-05, iter.max = 500,
                 im = TRUE, criteria = TRUE)
HeckmanEM.envelope(res, ylab="Normalized Quantile Residuals",xlab="Standard normal quantile",
                   line="quartile", col=c(20,1), pch=19, ylim = c(-5,4))
```

`HeckmanEM.infomat` *Standard error estimation for the Heckman Selectio model by the Information Matrix.*

Description

‘HeckmanEM.infomat()‘ estimate the standard errors for the parameters for the fitted Heckman selection model.

Usage

```
HeckmanEM.infomat(obj)
```

Arguments

`obj`	An object of the class HeckmanEM.

Value

The estimated standard errors for the parameters of the fitted model.

Examples

```
n <- 100
family <- "T"
nu <- 4
rho <- .6
cens <- .25

set.seed(20200527)
w <- cbind(1,runif(n,-1,1),rnorm(n))
x <- cbind(w[,1:2])
c <- qt(cens, df=nu)

sigma2 <- 1

beta <- c(1,0.5)
gamma <- c(1,0.3,-.5)
gamma[1] <- -c*sqrt(sigma2)

set.seed(1)
datas <- rHeckman(x,w,beta,gamma,sigma2,rho,nu,family=family)
y <- datas$y
cc <- datas$cc

res <- HeckmanEM(y, x, w, cc, nu = 4, family = "Normal", error = 1e-05, iter.max = 500,
                 im = FALSE, criteria = TRUE)
im <- HeckmanEM.infomat(res)
```

`rHeckman` *Generate data from the Heckman Selectio model (Normal or Student-t).*

Description

‘rHeckman()‘ generate a ramdom sample from the Heckman selection model (Normal or Student-t).

Usage

```
rHeckman(x, w, beta, gamma, sigma2, rho, nu = 4, family = "T")
```

Arguments

`x`	A covariate matrix for the response y.
`w`	A covariate matrix for the missing indicator cc.
`beta`	Values for the beta vector.
`gamma`	Values for the gamma vector.
`sigma2`	Value for the variance.

`rho`	Value for the dependence between the reponse and missing value.
`nu`	Value for the degrees of freedom.
`family`	The family to be used (Normal or T).

Value

Return an object with the response (y) and missing values (cc).

Examples

```
n <- 100
rho <- .6
cens <- 0.25
nu <- 4
set.seed(20200527)
w <- cbind(1,runif(n,-1,1),rnorm(n))
x <- cbind(w[,1:2])

family <- "T"
c <- qt(cens, df=nu)

sigma2 <- 1
beta <- c(1,0.5)
gamma<- c(1,0.3,-.5)
gamma[1] <- -c*sqrt(sigma2)

data <- rHeckman(x,w,beta,gamma,sigma2,rho,nu,family=family)
```

Index