

# Package ‘eive’

May 16, 2018

**Type** Package

**Title** An Algorithm for Reducing Errors-in-Variable Bias in Simple Linear Regression

**Version** 2.3

**Date** 2018-04-16

**Author** Mehmet Hakan Satman (PhD), Erkin Diyarbakirlioglu (PhD)

**Maintainer** Mehmet Hakan Satman <mhsatman@istanbul.edu.tr>

**Description** Performs a compact genetic algorithm search to reduce errors-in-variables bias in linear regression. The algorithm estimates the regression parameters with lower biases and higher variances but mean-square errors (MSEs) are reduced.

**License** GPL

**Imports** Rcpp (>= 0.11.1)

**LinkingTo** Rcpp

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2018-05-16 21:38:11 UTC

## R topics documented:

eive-package . . . . .	2
cga . . . . .	2
cga_generate_chromosome . . . . .	3
eive.cga . . . . .	4
generate.eive.data . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

 eive-package

*EIVE - Errors-in-Variable estimation*


---

### Description

This package includes functions for compact genetic algorithms and errors-in-variable estimation. The function 'eive' performs a genetic search to reduce the errors-in-variable bias in ordinary least squares estimator.

Change log: \* In version 2.1, more speed improvements by using lm.fit instead lm in critical code.  
 \* In version 2.0, some routines are rewritten in C++ and wrapped using Rcpp so a substantial speed improvement achieved.

### Details

Package: eive  
 Type: Package  
 Version: 2.1  
 Date: 2014-07-04  
 License: GPL

### Author(s)

Mehmet Hakan Satman <mhsatman@istanbul.edu.tr> Erkin Diyarbakirlioglu <ediyarbakirlioglu@gmail.com>  
 Maintainer: Mehmet Hakan Satman <mhsatman@istanbul.edu.tr>

---

 cga

*Function for performing compact genetic algorithms*


---

### Description

Function performs a compact genetic algorithm search for a given evaluation function.

### Usage

```
cga(chsize, popsize, evalFunc)
```

### Arguments

chsize	Number of bits
popsize	Number of population. By default it is 20
evalFunc	Function to minimize

**Details**

CGA (Compact genetic algorithms) sample chromosomes using this probability vector. A probability vector contains [P1,P2,...,PN] and the function generates and returns a chromosome [B1,B2,...,BN]. The probability of BK having the value of 1 is PK. So, it has more chance to have [1,1,1,0,0] than [0,0,0,1,1] when the probability vector is [0.9,0.9,0.9,0.1,0.1].

**Value**

Returns the best chromosome with size of chsize.

**Author(s)**

Mehmet Hakan Satman <mhsatman@istanbul.edu.tr> Erkin Diyarbakirlioglu <ediyarbakirlioglu@gmail.com>

---

cga\_generate\_chromosome  
*cga\_generate\_chromosome*

---

**Description**

Generates vector of zeros and ones for a given probability vector.

**Usage**

```
cga_generate_chromosome(prob_vec)
```

**Arguments**

prob\_vec            Vector of probabilities.

**Details**

This function is not directly called by user. CGA (Compact genetic algorithms) sample chromosomes using this probability vector. A probability vector contains [P1,P2,...,PN] and the function generates and returns a chromosome [B1,B2,...,BN]. The probability of BK having the value of 1 is PK. So, it has more chance to have [1,1,1,0,0] than [0,0,0,1,1] when the probability vector is [0.9,0.9,0.9,0.1,0.1].

**Value**

Returns the generated chromosome for a given probability vector. Return type is vector.

**Author(s)**

Mehmet Hakan Satman <mhsatman@istanbul.edu.tr> Erkin Diyarbakirlioglu <ediyarbakirlioglu@gmail.com>

---

eive.cga	<i>Errors-in-variable estimation in linear regression with compact genetic algorithms</i>
----------	---

---

### Description

Method performs a genetic search to find dummy variables that used in a two stage linear regression to reduce errors-in-variables bias in linear regression.

### Usage

```
eive.cga(dirtyx, otherx = NULL, y, numdummies = 10, popsize = 20)
```

### Arguments

dirtyx	Vector of values of independent variable measured with error
otherx	Matrix of other independent variables.
y	Vector of values of dependent variable
numdummies	Number of dummy variables used in algorithm. By default, it is 10.
popsize	Population size parameter used in CGA. By default it is 20.

### Details

Algorithm performs a genetic search to separate mismeasured independent variable into clean and error parts.

### Value

ols	lm object calculated using original values
eive	lm object calculated using the predicted variable by eive
proxy	lm object of proxy regression obtained by genetic search.

### Author(s)

Mehmet Hakan Satman <mhsatman@istanbul.edu.tr> Erkin Diyarbakirlioglu <ediyarbakirlioglu@gmail.com>

### Examples

```
# Creating an artificial data

# Loading required package
require("eive")

# Setting random number generator seed to 12345
# so each time the script runs, same numbers will
# be generated
set.seed(12345)
```

```

# Number of observations is set to 30
n<-30

# Unobserved X values are drawn from a Normal distribution
# with mean 10 and variance 7
clean.x <- rnorm(n, mean=10, sd=sqrt(7))

# Measurement error values are drawn from a Normal distribution
# with mean 0 and variance 3
delta.x <- rnorm(n, mean=0, sd=sqrt(3))

# Error term of regression. Normally distributed with mean 0 and
# variance 5
e <- rnorm(n, mean=0, sd=sqrt(5))

# Generating Y values using the linear model
# In this model, intercept is 20 and slope is 10.
y<- 20 + 10* clean.x + e

# Generating observed X values by adding measurement errors
# to unobserved X
dirty.x <- clean.x + delta.x

# Performs a genetic search to find dummy variables that
# used in two stage least squares.
# Please un-comment the line below
# result <- eive.cga (dirtyx=dirty.x, y=y, numdummies=10)

# Print the result
# Please un-comment the line below
# print(result)

##### OUTPUT #####
# $ols
#
# Call:
# lm(formula = y ~ dirtyx)
#
# Coefficients:
# (Intercept)      dirtyx
#    63.590      5.533
#
#
# $eive
#
# Call:
# lm(formula = y ~ ols.proxy$fitted.values)
#
# Coefficients:
# (Intercept)  ols.proxy$fitted.values
#    23.863      9.229
#

```

```

#
# $proxy
#
# Call:
# lm(formula = dirtyx ~ matrix(best, nrow = n))
#
# Coefficients:
#           (Intercept)  matrix(best, nrow = n)1  matrix(best, nrow = n)2
#           12.9321          -0.6252          -1.9923
# matrix(best, nrow = n)3  matrix(best, nrow = n)4  matrix(best, nrow = n)5
#           0.7537          -0.7076          -0.5247
# matrix(best, nrow = n)6  matrix(best, nrow = n)7  matrix(best, nrow = n)8
#           -0.9196          -2.0802          -0.9246
# matrix(best, nrow = n)9  matrix(best, nrow = n)10
#           -0.6164           1.9694
##### END OF OUTPUT #####

```

---

generate.eive.data      *Generates data for errors-in-variables model*

---

### Description

This function generates data using a linear regression model with size of  $n$ . Then one of the independent variables is contaminated by adding measurement errors. Another independent variable can be included in model.

### Usage

```
generate.eive.data(n, e.sd, delta.sd, seed = 12345, useotherx = FALSE)
```

### Arguments

<code>n</code>	Number of observations.
<code>e.sd</code>	Standard deviation of error term of regression.
<code>delta.sd</code>	Standard deviation of measurement error.
<code>seed</code>	Random number seed. By default, it is 12345.
<code>useotherx</code>	Boolean variable. If it is TRUE, another variable will be created with no errors. By default, it is FALSE.

### Value

Returns a matrix of contaminated variable, other variable (if exists) and independent variable in its columns.

### Author(s)

Mehmet Hakan Satman <mhsatman@istanbul.edu.tr> Erkin Diyarbakirlioglu <ediyarbakirlioglu@gmail.com>

# Index

cga, [2](#)  
cga\_generate\_chromosome, [3](#)  
  
eive (eive-package), [2](#)  
eive-package, [2](#)  
eive.cga, [4](#)  
  
generate.eive.data, [6](#)