Package ‘Monte.Carlo.se’

May 3, 2019

Type Package
Title Monte Carlo Standard Errors
Version 0.1.0
Author Dennis Boos, Kevin Matthew, Jason Osborne
Maintainer Dennis Boos <boos@ncsu.edu>
Description Computes Monte Carlo standard errors for summaries of Monte Carlo output. Sum-
maries and their standard errors are based on columns of Monte Carlo simulation output. Den-
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1
Suggests knitr, rmarkdown
VignetteBuilder knitr
NeedsCompilation no
Repository CRAN
Date/Publication 2019-05-03 10:30:04 UTC

R topics documented:

A.Quick.Outline ......................................................... 2
boot.se ................................................................. 2
Extra ................................................................. 4
jack.se ................................................................. 6
mc.se.matrix ......................................................... 9
mc.se.vector ....................................................... 11
pairwise.se ....................................................... 13
sim.samp ......................................................... 15

Index 17
A. Quick Outline

Monte.Carlo.se: A package for computing standard errors of Monte Carlo estimates

Description

The Monte.Carlo.se package has two main R functions `mc.se.vector` and `mc.se.matrix` that are built on two basic functions `jack.se` and `boot.se` for computing jackknife and bootstrap standard errors, respectively.

Details

A Monte Carlo Study often results in an N by k matrix X of estimates based on N independent samples of simulation data. X might contain parameter estimates (like the mean or median) or test results (say 0 for “accept the null hypothesis” or 1 for “reject the null”). Summary calculations from X, such as means or variances, are then displayed in a table or plot.

The purpose of the Monte.Carlo.se package is to compute estimates of the standard deviation (called standard errors) of these table entries.

Examples are included in the documentation for `mc.se.vector` and `mc.se.matrix`. However, more extended examples may be found in the three vignettes found by clicking “index” at the bottom of this page and then on ”User guides, package vignettes and other documentation.”

Author(s)

Dennis Boos, Kevin Matthew, Jason Osborne

boot.se

Bootstrap Standard Error

Description

`boot.se` – gives a bootstrap standard error (SE=estimated standard deviation)

Usage

`boot.se(x, B, theta, ...)`

Arguments

- `x` vector of data
- `B` number of bootstrap resamples (replications)
- `theta` function (statistic) applied to the data (e.g., mean, median, var)
- `...` Additional arguments to be passed
Details

The code was modified from code associated with the appendix of Efron and Tibshirani (1993)

Value

Returns the bootstrap SE of theta(x)

Author(s)

Dennis Boos, Kevin Matthew, Jason Osborne

References

Efron and Tibshirani (1993). _An Introduction to the Bootstrap_.

See Also

jack.se – mc.se.vector – mc.se.matrix

Examples

# simple example, data from Boos and Osborne (2105, Table 3)
# using theta=coefficient of variation mean/sd

x=c(1,2,7,9,5,17,11,2,15,85)
cv=function(x){sd(x)/mean(x)}

cv(x)
# [1] 1.383577

# bootstrap SE using B=1000

set.seed(384)
boot.se(x,B=1000,theta=cv)
# [1] 0.3416897

# More complex example using two samples, se for ratio of means
# data from Higgins (2003, problem 4.4, p. 142), LDH readings on 7 patients

before=c(89,90,87,98,120,85,97)
after=c(76,101,84,86,105,84,93)

# requires function using row index as "data", real data is extra parameter xdata
ratio.means <- function(index,xdata) 
{mean(xdata[index,1])/mean(xdata[index,2])}

ratio.means(index=1:7,xdata=data.frame(before,after))
# [1] 1.058824
# bootstrap SE for ratio of means

```r
set.seed(2917)
boot.se(x=1:7,B=1000,theta=ratio.means,xdata=data.frame(before,after))
# [1] 0.03576659
```

# To illustrate use with Monte Carlo output, first create some sample data
# 10,000 samples of size 15 from the Laplace (double exp) distribution

```r
NK<10000
set.seed(450)
z1 <- matrix(rexp(N*K*15),nrow=N)
z2 <- matrix(rexp(N*K*15),nrow=N)
z<-(z1-z2)/sqrt(2)  # subtract standard exponentials
out.m.15  <- apply(z,1,mean)
out.t20.15 <- apply(z,1,mean,trim=0.20)
out.med.15 <- apply(z,1,median)
```

# The three datasets (out.m.15,out.t20.15,out.med.15) each contain 10000 values.
# If we want use the variance of each column in a table, then to get
# the Monte Carlo standard error of those 3 variances,

```r
set.seed(250)
boot.se(out.m.15,B=1000,theta = var)
# [1] 0.0009373835

boot.se(out.t20.15,B=1000,theta = var)
# [1] 0.0007086057

boot.se(out.med.15,B=1000,theta = var)
# [1] 0.0008307258
# ends dontest
# Function Code

```root.se<-function(x, B, theta, ...){
call <- match.call()
n <- length(x)
bootsam <- matrix(sample(x, size = n * B, replace = T), nrow = B)
theta<apply(bootsam, 1, theta, ...)
se <- sd(theta)
return(se)
}
```

---

### Extra

#### Auxiliary Functions
**Description**

Auxiliary functions to be used in the Monte.Carlo.se package, mainly with mc.se.matrix. Scroll down to the Examples Section to see the actual code.

**Usage**

```r
ratio.var(index, xdata)
ratio.sd(index, xdata)
ratio.mse(index, xdata, true)
ratio.mean.what.var(index, xdata)
ratio.mean.sdhat.sd(index, xdata)
corr(index, xdata)
cv(x)
varn(x, n)
jack.var(x, theta, ...)
boot.var(x, B, theta, ...)
```

**Arguments**

- `index`: index = usually of the form 1:N
- `xdata`: actual data
- `true`: true parameter value when computing mean squared error
- `x`: Input vector in calls to jack.var and boot.var
- `n`: sample size
- `theta`: theta = function in calls to jack.var and boot.var
- `...`: Additional arguments to be passed
- `B`: Bootstrap reps in calls to boot.var

**Author(s)**

Dennis Boos, Kevin Matthew, Jason Osborne

**Examples**

```r
# These are extra functions included in the MCse package
# The following functions are to be used with mc.se.matrix

ratio.var <- function(index, xdata) # ratio of variances
```
\[
\frac{\text{var}(\text{xdata}[\text{index},1])}{\text{var}(\text{xdata}[\text{index},2])}
\]

# The above function is for the ratio of the sample variance of column 1 to
# the sample variance of column 2 of xdata.
# Note that the actual data goes into xdata, the second argument of \text{ratio.var}.
# Example call for 10,000 means and medians:
# \text{ratio.var}(1:10000, \text{xdata}=\text{cbind(\text{out.m.15}, \text{out.med.15})})

\text{ratio.sd}<\text{function(index, xdata)}{ \quad \# \text{ratio of standard deviations}
\text{sd}(\text{xdata}[\text{index},1])/\text{sd}(\text{xdata}[\text{index},2])}

\text{ratio.mse}<\text{function(index, xdata, true)}{ \quad \# \text{ratio of mean squared errors}
\text{mean}((\text{xdata}[\text{index},1]-\text{true})^2)/\text{mean}((\text{xdata}[\text{index},2]-\text{true})^2))}

\text{ratio.mean.what.var}<\text{function(index, xdata)}{ \# \text{estimates in col 1, what's in col 2}
\text{mean}(\text{xdata}[\text{index},1])/\text{var}(\text{xdata}[\text{index},2])}

\text{ratio.mean.sdhat.sd}<\text{function(index, xdata)}{ \# \text{estimates in col 1, SEs in col 2}
\text{mean}(\text{xdata}[\text{index},2])/\text{sd}(\text{xdata}[\text{index},1])}

\text{corr}<\text{function(index, xdata)}{ \quad \# \text{simple correlation}
\text{cor}(\text{xdata}[\text{index},1], \text{xdata}[\text{index},2])}

# These next two functions correspond to \text{jack.se} and \text{boot.se}.
# \text{x} is a data vector, and \text{theta} is a function applied to \text{x}.
# Each returns a variance estimate for \text{theta(x)}.

\text{jack.var} <- \text{function(x, theta, ...)}{ \# \text{jackknife estimate of variance}
\text{n} <- \text{length(x)}
\text{u} <- \text{rep(0, n)}
\text{for}(\text{i in 1:n})(\text{u[i]} <- \text{theta(x[ - i], ...)})
\text{jack.var} <- ((\text{n}-1)/\text{n})* \text{sum}(\text{(u-mean(u)}^2)
\text{return(jack.var))}

\text{boot.var} <- \text{function(x,B,theta, ...)}{ \# \text{bootstrap estimate of variance}
\text{n} <- \text{length(x)}
\text{bootsam} <- \text{matrix(sample(x, size = n*B, replace=T), nrow=B)}
\text{thetastar} <- \text{apply(bootsam, 1, theta,...)}
\text{boot.var} <- \text{var(thetastar)}
\text{return(boot.var))}

---

\text{jack.se} \hspace{1cm} \text{Jackknife Standard Error}

**Description**

\text{jack.se} — gives jackknife standard error (SE=estimated standard deviation)
Usage

closest.se(x, theta, ...)

Arguments

x vector of data
theta — function (statistic) applied to the data (e.g., mean, median, var)
... Additional arguments to be passed

Details

The code was modified from code associated with the appendix of Efron and Tibshirani (1993)

Value

Returns the jackknife standard error of theta(x)

Author(s)

Dennis Boos, Kevin Matthew, Jason Osborne

References

Efron and Tibshirani (1993), "An Introduction to the Bootstrap".

See Also

boot.se — mc.se.vector — mc.se.matrix

Examples

# simple example, data from Boos and Osborne (2015, Table 3)
# using theta = coefficient of variation = mean/sd
x=c(1,2,79,5,17,11,2,15,85)
cv=function(x){sd(x)/mean(x)}
cv(x)
# [1] 1.383577

closest.se(x,theta=cv)
# [1] 0.3435321

# More complex example using two samples, se for ratio of means
# data from Higgins (2003, problem 4.4, p. 142), LDH readings on 7 patients
before=c(89,90,87,98,120,85,97)
after=c(76,101,84,86,105,84,93)
# requires function using row index as "data,"
# real data is extra parameter xdata

c\text{ratio\_means} \leftarrow \text{function}(\text{index}, \text{xdata})\{
  \text{mean}(\text{xdata}[\text{index}, 1]) / \text{mean}(\text{xdata}[\text{index}, 2])\}

# ratio of means for before-after data

c\text{ratio\_means}(\text{index}=1:7, \text{xdata}\leftarrow \text{data\_frame}(\text{before, after}))
# [1] 1.058824

# jackknife SE for ratio of means

c\text{jack\_se}(\text{x}=1:7, \theta=\text{ratio\_means}, \text{xdata}\leftarrow \text{data\_frame}(\text{before, after}))
# [1] 0.03913484

# To illustrate use with Monte Carlo output, first create some sample data
# 10,000 samples of size 15 from the Laplace (double exp) distribution

\text{N}<\text{10000}
\text{set\_seed}(450)
\text{z1} \leftarrow \text{matrix}(\text{rexp}(\text{N}\times15), \text{nrow} = \text{N})
\text{z2} \leftarrow \text{matrix}(\text{rexp}(\text{N}\times15), \text{nrow} = \text{N})
\text{z} \leftarrow (\text{z1} \times \text{z2}) / \text{sqrt}(2)  \quad \# \text{subtract standard exponentials}
\text{out\_m.15} \leftarrow \text{apply}(\text{z}, 1, \text{mean})
\text{out\_t.20.15} \leftarrow \text{apply}(\text{z}, 1, \text{mean}, \text{trim}=0.20)
\text{out\_med.15} \leftarrow \text{apply}(\text{z}, 1, \text{median})

# The three datasets (out\_m.15, out\_t.20.15, out\_med.15) each contain 10,000 values.
# If we want use the variance of each column in a table, then to get
# the Monte Carlo standard error of those 3 variances,

c\text{jack\_se}(\text{out\_m.15}, \theta = \text{var})
# [1] 0.0009612314

c\text{jack\_se}(\text{out\_t.20.15}, \theta = \text{var})
# [1] 0.0007008859

c\text{jack\_se}(\text{out\_med.15}, \theta = \text{var})
# [1] 0.0008130531

# Function Code

c\text{jack\_se=}\text{function}(\text{x, theta, ...}{(}
  \text{call} \leftarrow \text{match\_call}()
  \text{n} \leftarrow \text{length}(\text{x})
  \text{u} \leftarrow \text{rep}(0, \text{n})
  \text{for(} i \text{ in } 1:\text{n} \text{) } \{\text{u}[i] \leftarrow \text{theta}(\text{x}[ - i], ...)\}
  \text{jack\_se} \leftarrow \text{sqrt}(((\text{n} - 1)/\text{n}) \times \text{sum}((\text{u} - \text{mean}(\text{u}))^2))
  \text{return}(\text{jack\_se})}
**mc.se.matrix**  
*Standard Errors for Summaries Based on 2 or More Vectors of Monte Carlo Output*

**Description**

`mc.se.matrix` — jackknife and bootstrap SEs for statistics based on k correlated samples.

**Usage**

`mc.se.matrix(x, xcol, B = 0, seed = NULL, summary.f, ...)`

**Arguments**

- `x`: N by k matrix or data frame of MC output, k>1
- `xcol`: vector specifying columns of x to use
- `B`: B=0 (default) means use jackknife, B>0 means use bootstrap with B resamples. If B>0, then a seed must be given to start the bootstrap resampling.
- `seed`: seed=NULL (default) used with jackknife, otherwise needs positive integer
- `summary.f`: summary function with arguments (index,xdata), x goes into xdata
- `...`: Additional arguments to be passed

**Details**

Suppose that k columns of Monte Carlo output are produced from N independent Monte Carlo samples, and summary statistics involving 2 or more columns are to be reported in tables. `mc.se.matrix` gives Monte Carlo standard errors (SEs) for these summary statistics (e.g., the ratio of 2 column means or variances). The vignette `vignette("Example2", package = "Monte.Carbo.se")` is a detailed account of using `mc.se.vector`.

**Value**

Returns data frame of summary.f, Monte Carlo (MC) SE of summary.f, MC sample size N, method (jackknife or bootstrap), B and seed if bootstrap is used.

**Author(s)**

Dennis Boos, Kevin Matthew, Jason Osborne

**References**

Examples

# First create MC output used in Table 9.1, p. 367, of Boos and Stefanski (2013).
# norm15 holds 10,000 sample means, 20% trimmed means, and medians
# for normal samples of size 15

N <- 10000
set.seed(346)  # sets the random number seed
z <- matrix(rnorm(N*15), nrow=N)  # N rows of N(0,1) samples, n=15
out.m.15 <- apply(z, 1, mean)  # mean for each sample
out.t20.15 <- apply(z, 1, mean, trim=0.20)  # 20% trimmed mean for each sample
out.med.15 <- apply(z, 1, median)  # median for each sample

# Save all 10000 blocks of 3 estimators in a data frame
norm15 <- data.frame(mean=out.m.15, trim20=out.t20.15, median=out.med.15)

# Pearson correlation based on 2 vectors of MC output
# Note that the 2 vectors are in xdata, index is for the rows of xdata
corr <- function(index, xdata){cor(xdata[index,1], xdata[index,2])}

# Compute jackknife SE for summary.f=corr
mc.se.matrix(norm15, xcol=c(1,2), summary.f=corr)
# summary se N method
# 1 0.9367602 0.001256079 10000 Jackknife

# Compute bootstrap SE for summary.f=corr
mc.se.matrix(norm15, xcol=c(1,2), summary.f=corr, B=10000, seed=3928)
# summary se N method B seed
# 1 0.9367602 0.001287065 10000 Bootstrap 1000 3928

# Rerun with B=5000
mc.se.matrix(norm15, xcol=c(1,2), summary.f=corr, B=5000, seed=3928)
# summary se N method B seed
# 1 0.9367602 0.001266177 10000 Bootstrap 5000 3928

# Compute jackknife SE for summary.f=ratio.var
# = ratio of variances of the two columns
# A ratio of 2 variances facilitates comparison of the variances
ratio.var <- function(index, xdata)
{var(xdata[index,1])/var(xdata[index,2])}

# ratio of column 1 variance to column 2 variance
mc.se.matrix(norm15, xcol=c(1,2), summary.f=ratio.var)
# summary se N method
# 1 0.8895367 0.006263652 10000 Jackknife
# Coupled with SE=0.006, the ratio=0.89 shows the second variance is larger than the first
mc.se.vector Standard Errors for Monte Carlo Output Summaries

Description

mc.se.vector — gives jackknife and bootstrap SEs for a vector of MC output (say N estimates)

Usage

mc.se.vector(x, summary.f, B = 0, seed = NULL, ...)

Arguments

x vector from N independent Monte Carlo replications
summary.f summary function computed from x (e.g., mean, median, var)
B B=0 means use jackknife (default), B>0 means use bootstrap with B resamples, If B>0, then a seed must be given to start the bootstrap resampling
seed seed=NULL (default) used with jackknife, otherwise needs a positive integer
... Additional arguments to be passed
Details

Suppose that an N-vector of Monte Carlo output (thus, a sample of size N) is produced from N independent Monte Carlo samples, and a summary statistic like the mean or variance is to be reported in a table. `mc.se.vector` gives Monte Carlo standard errors (SEs) for these summary statistics. The vignette vignette("Example1", package = "Monte.Carlo.se") is a detailed account of using `mc.se.vector`.

Value

Returns data frame of summary.f, MC standard error of summary.f, MC sample size N, method (jackknife or bootstrap), B and seed if bootstrap is used.

Author(s)

Dennis Boos, Kevin Matthew, Jason Osborne

References


Examples

# First create MC output used for Table 9.1, p. 367, of Boos and Stefanski (2013)
# norm15 holds 10,000 sample means, 20% trimmed means, and medians
# computed from normal samples of size 15

N <- 10000
set.seed(346)  # sets the random number seed
z <- matrix(rnorm(N*15), nrow=N)  # N rows of N(0,1) samples, n=15
out.m.15 <- apply(z,1,mean)  # mean for each sample
out.t20.15 <- apply(z,1,mean,trim=0.20)  # 20% trimmed mean for each sample
out.med.15 <- apply(z,1,median)  # median for each sample

# Save all 1000 blocks of 3 estimators in a data frame
norm15 <- data.frame(mean=out.m.15,trim20=out.t20.15,median=out.med.15)

# Compute the jackknife Standard Error (SE) for summary.f=mean
# This summary is useful for estimating the bias of estimators.
mc.se.vector(norm15[,1],summary.f=mean)
# summary se N method
# 1 -0.001920642 0.00256714 10000 Jackknife

# Compute a bootstrap SE for summary.f=mean
mc.se.vector(norm15[,1],B=1000,seed=4822,summary.f=mean)
# summary se n method B seed
# 1 -0.001920642 0.002573516 10000 Bootstrap 1000 4822
# compare to basic R

```r
mean(norm15[,1])
# [1] -0.001920642
```

```r
sd(norm15[,1])/sqrt(10000)
# [1] 0.00256714
```

# Illustrate use with summaries having additional parameters.
# Compute the jackknife SE for summary.f=varn
# Multiplying the variance by sample size n allows comparison for different n

```r
varn=function(x,n){n*var(x)}
```

# The additional parameter n replaces the ... in the mc.se.vector definition

```r
mc.se.vector(norm15[,1],summary.f=varn,n=15)
# summary se N method
# 1 0.9885314 0.01407249 10000 Jackknife
```

# Compute a bootstrap SE for summary.f=varn

```r
mc.se.vector(norm15[,1],B=1000,seed=3029,summary.f=varn,n=15)
# summary se n method B seed
# 1 0.9885314 0.01408173 10000 Bootstrap 1000 3029
```

# Function Code

```r
mc.se.vector <- function(x,summary.f,B=0,seed=NULL,...){
  # x is a vector from N independent Monte Carlo replications
  # summary.f is a summary function computed from MC output
  # ... is for additional arguments to summary.f
  # B=0 means use jackknife, B>0 means use bootstrap with B resamples
  if (B==0){
    if(is.null(seed))stop('If B>0, then seed for the bootstrap must be given in the call')
    set.seed(seed);se=boot.se(x,B,B,theta=summary.f,...)
  }
  else{se=jack.se(x,theta=summary.f,...)}
  summ = summary.f(x,...)
  if(B>0){out=data.frame(summary=summ,se,n=length(x),method="Bootstrap",B,seed)}
  else{out=data.frame(summary=summ,se,N=length(x),method="Jackknife")}
  return(out)}
```

---

pairwise.se Standard Errors for Paired Comparisons of Monte Carlo Output Summaries
Description

pairwise.se — gives jackknife and bootstrap SEs for all the pairwise difference or ratios of Monte Carlo summaries.

Usage

pairwise.se(x, xcol, diff = TRUE, digits = 4, B = 0, seed = NULL, summary.f, ...)

Arguments

x vector from N independent Monte Carlo replications
xcol columns of x to be used
diff If TRUE (default), uses differences; if diff=F, uses ratios
digits number of digits to retain in output data frame
B B=0 means use jackknife (default), B>0 means use bootstrap with B resamples, If B>0, then a seed must be given to start the bootstrap resampling
seed seed=NULL (default) used with jackknife, otherwise needs a positive integer
summary.f summary function computed from x (e.g., mean, median, var)
... Additional arguments to be passed

Details

Suppose that an N-vector of Monte Carlo output (thus, a sample of size N) is produced from N independent Monte Carlo samples, and a summary statistic like the mean or variance is to be reported in a table. pairwise.se gives Monte Carlo standard errors (SEs) for all pairwise differences or ratios of these summary statistics. The vignette vignette("Example3", package = "Monte.Carlo.se") is a detailed account of using pairwise.se.

Value

Returns a data frame of the individual ith and jth column summaries (summi and summj), the differences or ratios of those summaries (summary), MC standard error of the difference or ratio, MC sample size N, method (jackknife or bootstrap), B and seed if bootstrap is used

Author(s)

Dennis Boos, Kevin Matthew, Jason Osborne

Examples

# Using the output data matrix hold generated in vignette Example3,
# calculate jackknife and bootstrap standard errors
# for the differences and ratios of the CV estimates.

# Jackknife SE of Differences of CVs
sim.samp

Code for Generating Random Samples

Description

sim.samp — generates N samples of size n from distribution DIST

Usage

sim.samp(nrep, n, DIST, ...)

Arguments

nrep = number of random data sets
n = sample size of each data set
DIST = distribution to generate from, e.g., runif, rnorm, additional parameters in ...
... Additional arguments to be passed
Value

N by n matrix, each row is a data set of size n

Author(s)

Dennis Boos, Kevin Matthew, Jason Osborne

Examples

N<-100
set.seed(346)     # sets the random number seed
sim.samp(N,15,rnorm)->z # 100 N(0,1) samples of size n=15
sim.samp(N,40,rnorm,mean=10,sd=5)->z # 100 N(10,25) samples of size n=40

# Function Code

sim.samp <- function(nrep,n,DIST,...){
  data <- matrix(DIST(n * nrep, ...), ncol = n, nrow = nrep)}
Index

A. Quick Outline, 2

boot.se, 2, 2, 7
boot.var (Extra), 4

corr (Extra), 4
cv (Extra), 4

Extra, 4

jack.se, 2, 3, 6
jack.var (Extra), 4

mc.se.matrix, 2, 3, 7, 9
mc.se.vector, 2, 3, 7, 11

pairwise.se, 13

ratio.mean.sdhat.sd (Extra), 4
ratio.mean.vhat.var (Extra), 4
ratio.mse (Extra), 4
ratio.sd (Extra), 4
ratio.var (Extra), 4

sim.samp, 15

varn (Extra), 4