Package 'edcc'

February 19, 2015

Version 1.0-0

Title Economic Design of Control Charts

Author Weicheng Zhu, Changsoon Park

Maintainer Weicheng Zhu <weicheng@dreamhunter.me>

Depends spc

Description This package provides a unified approach for Economic Design of Control Charts. The main purpose of this package is to find out the optimal parameters to minimize the ECH (Expected Cost per Hour) of the process.

License GPL

Collate 'ecoDesign.R' 'internal.R'

Repository CRAN

Date/Publication 2013-01-18 07:17:47

NeedsCompilation no

R topics documented:

`contour.edcc` *Contour plot of an "edcc" class object*

Description

contour plot of an "edcc" class object

Usage

```
## S3 method for class 'edcc'
contour(x, call.print = TRUE, ...)
```

Arguments

`x`	an object of "edcc" class
`call.print`	a logical value indicating whether the the R command should be printed on the contour plot. Default is TRUE
`...`	arguments to be passed to contour plot, see `contour` for details

Details

S3 method of contour plot for "edcc" class object

Value

a contour plot

See Also

`ecoXbar`, `ecoCusum`, `ecoEwma`, `update.edcc`, `contour`

Examples

```
x <- ecoXbar(h=seq(0.7,0.9,by=.01),L=seq(2.8,3.3,by=.01),n=4:6,P0=110,
P1=10,nlevels=50,contour.plot=TRUE)
contour(x,nlevels=20,lty=2,col=2,call.print=FALSE)
```

ecoCusum *Economic design for the CUSUM control chart*

Description

Calculate the optimum parameters of n(sample size), h(sampling interval), k(reference value) and H(decision interval) for Economic Design of the CUSUM control chart. For more information about the reference value see 'Details'.

Usage

```
ecoCusum(h, H, n, delta = 2, lambda = 0.01, P0 = NULL,
  P1 = NULL, C0 = NULL, C1 = NULL, Cr = 20, Cf = 10,
  T0 = 0, Tc = 0.1, Tf = 0.1, Tr = 0.2, a = 0.5, b = 0.1,
  d1 = 1, d2 = 1, nlevels = 30, sided = "one",
  par = NULL, contour.plot = FALSE, call.print = TRUE,
  ...)

echCusum(h, H, n, delta = 2, lambda = 0.01, P0 = NULL,
  P1 = NULL, C0 = NULL, C1 = NULL, Cr = 20, Cf = 10,
  T0 = 0, Tc = 0.1, Tf = 0.1, Tr = 0.2, a = 0.5, b = 0.1,
  d1 = 1, d2 = 1, sided = "one")
```

Arguments

h	sampling interval. It can be a numeric vector or left undefined. See 'Details'
H	decision interval. It can be a numeric vector or left undefined. See 'Details'
n	sample size. It can be an integer vector or left undefined. See 'Details'
delta	shift in process mean in standard deviation units when assignable cause occurs (delta = |mu1 - mu0|/sigma), where sigma is the standard deviation of observations; mu0 is the in-control process mean; mu1 is the out-of-control process mean. Default value is 2.
lambda	we assume the in-control time follows a exponential distribution with mean 1/lambda. Default value is 0.05.
P0	profit per hour earned by the process operating in control. See 'Details'.
P1	profit per hour earned by the process operating out of control
C0	cost per hour due to nonconformities produced while the process is in control.
C1	cost per hour due to nonconformities produced while the process is out of control.(C1 > C0)
Cr	cost for searching and repairing the assignable cause, including any downtime.
Cf	cost per false alarm, including the cost of searching for the cause and the cost of downtime if production ceases during search.
T0	time to sample and chart one item.

`Tc`	expected time to discover the assignable cause.
`Tf`	expected search time when false alarm occurs.
`Tr`	expected time to repair the process.
`a`	fixed cost per sample.
`b`	cost per unit sampled.
`d1`	flag for whether production continues during searches (1-yes, 0-no). Default value is 1.
`d2`	flag for whether production continues during repairs (1-yes, 0-no). Default value is 1.
`nlevels`	30. It works only when `contour.plot` is TRUE.
`sided`	distinguish between one-, two-sided and Crosier's modified two-sided CUSUM scheme by choosing "one", "two", and "Crosier", respectively. See details in `xcusum.arl`
`par`	initial values for the parameters to be optimized over. It can be a vector of length 2 or 3. See 'Details'
`contour.plot`	a logical value indicating wether a contour plot should be drawn. Default is FALSE.
`call.print`	a logical value indicating whether the "call" should be printed on the contour plot. Default is TRUE
`...`	other arguments to be passed to `optim` function.

Details

When parameter `par` is specified, optimization algorithms would be used as default. `par` can be specified as: `par = c(h, H)` where `h` and `H` are the initial values of smapling interval and decision interval when `n` is specified; or `par = c(h, H, n)`. Good inital values may lead to good optimum results.

When parameters `h`, `H`, `n` are all undefined, `ecoCusum` function will try to find the global optimum point to minimize the ECH (Expected Cost per Hour) using optimization algorithms (`optim` function), but in this case `n` would not be integer. It is usually helpful for the experimenter to find the region where the optimum point may exist quickly. When `h` and `H` are undefined but `n` is given as an integer vector, `ecoCusum` function will try to find the optimum point for each `n` value using optimization algorithms. When `h`, `H` and `n` are all given, `ecoCusum` function will use a "grid method" way to calculate the optimum point, that is ECH for all the combinations of the parameters will be calculated. The "grid method" way is much slower than using optimization algorithms, but it would be a good choice when optimization algorithms fail to work well.

There is strong numerical and theoretical evidence that for given L1, the value of L0 approaches its maximum when k(reference value) is chosen mid-way the between AQL and the RQL: $k = mu0 + 0.5*delta*sigma (Appl. Statist.(1974) 23, No. 3, p. 420). For this reason we treat k as a constant value and optimize n, h and H. For cost parameters either P0, P1 or C0, C1 is needed. If P0 and P1 are given, they will be used first, else C0 and C1 will be used. For economic design of the CUSUM chart, when `d1` and `d2` are both 1, only if the difference between P0 and P1 keeps the same, the results are identical. If the difference between C0 and C1 keeps the same, the optimum parameters are almost the same but the ECH(Expected Cost per Hour) values will change.

`echCusum` is used to calculate the ECH (Expected Cost per Hour) for one given design point.

Value

The ecoCusum function returns an object of class "edcc", which is a list of elements optimum, cost.frame, FAR and ATS. optimum is a vector with the optimum parameters and the corresponding ECH value; cost.frame is a dataframe with the optimum parameters and the corresponding ECH values for all given n(if n is not specified, cost.frame won't be returned); FAR indicates the false alarm rate during the in-control time, which is calculated as lambda*(average number of false alarm); ATS indicates the average time to signal after the occurrence of an assignable cause, calculated as h*ARL2 - tau, where tau is the expected time of occurrence of the assignable cause given it occurs between the i-th and (i+1)st samples. The echCusum function returns the calculated ECH value only.

References

Weicheng Zhu, Changsoon Park (2013), edcc: An R Package for the Economic Design of the Control Chart. *Journal of Statistical Software*, 52(9), 1-24. http://www.jstatsoft.org/v52/i09/

Lorenzen and Vance (1986). The economic design of control charts: a unified approach, *Technometrics*, 28. 3-10.

Chiu, W.K. (1974). The economic design of CUSUM charts for controlling normal means, *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 23(3), 420-433.

See Also

ecoXbar, ecoEwma, xcusum.arl, optim, update.edcc, contour

Examples

```
#Chiu, W.K. (1974). Applied Statistics, 23, p427 Table3, row 1-2,14
## LINE 1
## global optimization to h, H and n, when lambda = 0.01, "Nelder-Mead" optimization algorithm doesn't work
#(y <- ecoCusum( P0=150,P1=50,Cr=30,d1=0,d2=0))
## we can try other algorithms:
(y1 <- ecoCusum( P0=150,P1=50,Cr=30,d1=0,d2=0,method="BFGS"))
# Based on the global optimum above, we specify the range of the
# parameters like this
(yy1 <- ecoCusum( h=seq(1.3,1.45,by=.01), H=seq(.5,0.6,by=.01),n=4:6,
P0=150,P1=50,Cr=30,d1=0,d2=0))
## LINE 2
(y2 <- ecoCusum( P0=150,P1=50,Cr=30,d1=0,d2=0,lambda=0.05))
(yy2 <- ecoCusum( h=seq(.6,0.7,by=.01), H=seq(.5,0.6,by=.01),n=3:6,
P0=150,P1=50,Cr=30,d1=0,d2=0,lambda=0.05))
contour(yy2)
## LINE 14
(y14 <- ecoCusum(n=30,P0=150,P1=50,Cr=30,delta=0.5,d1=0,d2=0,method="L-BFGS-B"))
(yy14 <- ecoCusum(h=seq(2.55,2.65,by=0.01),H=seq(0.3,0.4,by=0.01),
n=28:30,P0=150,P1=50,Cr=30,delta=0.5,d1=0,d2=0))
#Douglas (2009). Statistical quality control: a modern introduction, sixth edition, p470.
ecoCusum(lambda=.05,P0=110,P1=10,Cr=25,Cf=50,Tr=0,Tf=0,Tc=1,T0=.0167,a=1)
ecoCusum(h=seq(0.75,0.85,by=.01),H=seq(.55,0.65,by=.01),n=4:6,lambda=.05,
P0=110,P1=10,Cr=25,Cf=50,Tr=0,Tf=0,Tc=1,T0=.0167,a=1)
```

ecoEwma	*Economic design for the EWMA control chart*

Description

Calculate the optimum parameters, n(sample size), h(sampling interval), w(weight to the present sample) and k(number of s.d. from control limits to center line) for economic Design of the EWMA control chart .

Usage

```
ecoEwma(h, w, k, n, delta = 2, lambda = 0.05, P0 = NULL,
  P1 = NULL, C0 = NULL, C1 = NULL, Cr = 25, Cf = 10,
  T0 = 0.0167, Tc = 1, Tf = 0, Tr = 0, a = 1, b = 0.1,
  d1 = 1, d2 = 1, nlevels = 30, sided = "two",
  par = NULL, contour.plot = FALSE, call.print = TRUE,
  ...)

echEwma(h, w, k, n, delta = 2, lambda = 0.05, P0 = NULL,
  P1 = NULL, C0 = NULL, C1 = NULL, Cr = 25, Cf = 10,
  T0 = 0.0167, Tc = 1, Tf = 0, Tr = 0, a = 1, b = 0.1,
  d1 = 1, d2 = 1, sided = "two")
```

Arguments

h	sampling interval. It can be a numeric vector or left undefined. See 'Details'
w	the weight value between 0 and 1 given to the latest sample. It must be specified.
k	control limit coefficient. It can be a numeric vector or left undefined. See 'Details'
n	sample size. It can be an integer vector or left undefined. See 'Details'
delta	shift in process mean in standard deviation units when assignable cause occurs (delta = |mu1 - mu0|/sigma), where sigma is the standard deviation of observations; mu0 is the in-control process mean; mu1 is the out-of-control process mean. Default value is 2.
lambda	we assume the in-control time follows a exponential distribution with mean 1/lambda. Default value is 0.05.
P0	profit per hour earned by the process operating in control. See 'Details'.
P1	profit per hour earned by the process operating out of control
C0	cost per hour due to nonconformities produced while the process is in control.
C1	cost per hour due to nonconformities produced while the process is out of control.(C1 > C0)
Cr	cost for searching and repairing the assignable cause, including any downtime.
Cf	cost per false alarm, including the cost of searching for the cause and the cost of downtime if production ceases during search.

`T0`	time to sample and chart one item.
`Tc`	expected time to discover the assignable cause.
`Tf`	expected search time when false alarm occurs.
`Tr`	expected time to repair the process.
`a`	fixed cost per sample.
`b`	cost per unit sampled.
`d1`	flag for whether production continues during searches (1-yes, 0-no). Default value is 1.
`d2`	flag for whether production continues during repairs (1-yes, 0-no). Default value is 1.
`nlevels`	number of contour levels desired. Default value is 30. It works only when `contour.plot` is TRUE.
`sided`	distinguish between one- and two-sided EWMA control chart by choosing "one" and "two", respectively. See details in `xewma.arl`
`par`	initial values for the parameters to be optimized over. It can be a vector of length 2 or 3. See 'Details'
`contour.plot`	a logical value indicating whether a contour plot should be drawn. Default is FALSE.
`call.print`	a logical value indicating whether the "call" should be printed on the contour plot. Default is TRUE
`...`	other arguments to be passed to contour function.

Details

Parameter `w` should always be given, because the range of `w` is so restricted that optimization algorithms usually don't converge.

When parameter `par` is specified, optimization algorithms would be used as default. `par` can be specified as: `par = c(h, k)` where `h` and `k` are the initial values of smapling interval and control limit when `n` is specified; or `par =  c(h, k, n)`. Good inital values may lead to good optimum results.

When parameters `h`, `k`, `n` are all undefined, `ecoEwma` function will try to find the global optimum point to minimize the ECH (Expected Cost per Hour) using optimization algorithms (`optim` function), but in this case `n` would not be integer. It is usually helpful for the experimenter to find the region where the optimum point may exist quickly. When `h` and `k` are undefined but `n` is given as an integer vector, `ecoEwma` function will try to find the optimum point for each `n` value using optimization algorithms. When `h`, `k` and `n` are all given, `ecoEwma` function will use a "grid method" way to calculate the optimum point, that is ECH for all the combinations of the parameters will be calculated. The "grid method" way is much slower than using optimization algorithms, but it would be a good choice when optimization algorithms fail to work well.

For cost parameters either P0, P1 or C0, C1 is needed. If P0 and P1 are given, they will be used first, else C0 and C1 will be used. For economic design of the EWMA chart, when `d1` and `d2` are both 1, only if the difference between P0 and P1 keeps the same, the results are identical. If the difference between C0 and C1 keeps the same, the optimum parameters are almost the same but the ECH(Expected Cost per Hour) values will change.

`echEwma` is used to calculate the ECH (Expected Cost per Hour) for one given design point.

Value

The ecoEwma function returns an object of class "edcc", which is a list of elements optimum, cost.frame, FAR and ATS. optimum is a vector with the optimum parameters and the corresponding ECH value; cost.frame is a dataframe with the optimum parameters and the corresponding ECH values for all given n(if n is not specified, cost.frame won't be returned); FAR indicates the false alarm rate during the in-control time, which is calculated as lambda*(average number of false alarm); ATS indicates the average time to signal after the occurrence of an assignable cause, calculated as h*ARL2 - tau, where tau is the expected time of occurrence of the assignable cause given it occurs between the i-th and (i+1)st samples. The echEwma function returns the calculated ECH value only.

References

Weicheng Zhu, Changsoon Park (2013), edcc: An R Package for the Economic Design of the Control Chart. *Journal of Statistical Software*, 52(9), 1-24. http://www.jstatsoft.org/v52/i09/

Lorenzen and Vance (1986). The economic design of control charts: a unified approach, *Technometrics*, 28. 3-10.

See Also

ecoXbar, ecoCusum, xewma.arl, update.edcc, optim,contour

Examples

```
#Douglas (2009). Statistical quality control: a modern introduction, sixth edition, p470.
## Set w from 0.1 to 1 by 0.1 to catch the trend.
ecoEwma(w=seq(0.1,1,by=0.1),P0=110,P1=10,Cf=50)
#yy = ecoEwma(h = seq(.7,1,by=.01), w = seq(0.8,1,by=.01),k = seq(2.9,3.3, by = 0.01), n = 4:5, P0 = 110, P1 = 10,

##$optimum
##Optimum h Optimum k Optimum n Optimum w       ECH
##  0.81000   2.99000   5.00000   0.95000  10.36482
#contour(yy)
```

---

ecoXbar *Economic design for the X-bar control chart*

Description

Calculate the optimum parameters, n(sample size), h(sampling interval) and L(number of s.d. from control limits to center line) for Economic Design of the X-bar control chart .

Usage

```
ecoXbar(h, L, n, delta = 2, lambda = 0.05, P0 = NULL,
  P1 = NULL, C0 = NULL, C1 = NULL, Cr = 25, Cf = 50,
  T0 = 0.0167, Tc = 1, Tf = 0, Tr = 0, a = 1, b = 0.1,
  d1 = 1, d2 = 1, nlevels = 30, sided = "two",
  par = NULL, contour.plot = FALSE, call.print = TRUE,
  ...)

echXbar(h, L, n, delta = 2, lambda = 0.05, P0 = NULL,
  P1 = NULL, C0 = NULL, C1 = NULL, Cr = 25, Cf = 50,
  T0 = 0.0167, Tc = 1, Tf = 0, Tr = 0, a = 1, b = 0.1,
  d1 = 1, d2 = 1, sided = "two")
```

Arguments

h	sampling interval. It can be a numeric vector or left undefined. See 'Details'
L	number of standard deviations from control limits to center line. It can be a numeric vector or left undefined. See 'Details'
n	sample size. It can be an integer vector or left undefined. See 'Details'
lambda	we assume the in-control time follows an exponential distribution with mean 1/lambda. Default value is 0.05.
delta	shift in process mean in standard deviation units when assignable cause occurs (delta = (mu1 - mu0)/sigma), where sigma is the standard deviation of observations; mu0 is the in-control process mean; mu1 is the out-of-control process mean. Default value is 2.
P0	profit per hour earned by the process operating in control. See 'Details'.
P1	profit per hour earned by the process operating out of control(P0 > P1).
C0	cost per hour due to nonconformities produced while the process is in control.
C1	cost per hour due to nonconformities produced while the process is out of control.(C1 > C0)
Cr	cost for searching and repairing the assignable cause, including any downtime.
Cf	cost per false alarm, including the cost of searching for the cause and the cost of downtime if production ceases during search.
T0	time to sample and chart one item.
Tc	expected time to discover the assignable cause.
Tf	expected search time when false alarm occurs.
Tr	expected time to repair the process.
a	fixed cost per sample.
b	cost per unit sampled.
d1	flag for whether production continues during searches (1-yes, 0-no). Default value is 1.
d2	flag for whether production continues during repairs (1-yes, 0-no). Default value is 1.

nlevels	number of contour levels desired. Default value is 30. It works only when `contour.plot` is TRUE.
sided	distinguish between one- and two-sided X-bar chart by choosing “one” or “two” respectively. When `sided = ``one''`, `delta > 0` means the control chart for detecting a positive shift, and vice versa. Default is “two”.
par	initial values for the parameters to be optimized over. It can be a vector of length 2 or 3. See 'Details'
contour.plot	a logical value indicating whether a contour plot should be drawn. Default is FALSE. Only works when the parameters h, L and n are all specified.
call.print	a logical value indicating whether the "call" should be printed on the contour plot. Default is TRUE
...	other arguments to be passed to `optim` function.

Details

When parameter `par` is specified, optimization algorithms would be used as default. `par` can be specified as: `par = c(h, L)` where `h` and `L` are the initial values of smapling interval and control limit when `n` is specified; or `par = c(h, L, n)`. Good inital values may lead to good optimum results.

When parameters `h`, `L`, `n` are all undefined, `ecoXbar` function will try to find the global optimum point to minimize the ECH (Expected Cost per Hour) using optimization algorithms (`optim` function), but in this case `n` would not be integer. It is usually helpful for the experimenter to find the region where the optimum point may exist quickly. When `h` and `L` are undefined but `n` is given as an integer vector, `ecoXbar` function will try to find the optimum point for each `n` value using optimization algorithms. When `h`, `L` and `n` are all given, `ecoXbar` function will use the "grid method" to calculate the optimum point, that is ECH for all the combinations of the parameters will be calculated. The "grid method" way is much slower than using optimization algorithms, but it would be a good choice when optimization algorithms fail to work well.

For cost parameters either P0, P1 or C0, C1 is needed. If P0 and P1 are given, they will be used first, else C0 and C1 will be used. For economic design of the X-bar chart, when `d1` and `d2` are both 1, only if the difference between P0 and P1 keeps the same, the results are identical. If the difference between C0 and C1 keeps the same, the optimum parameters are almost the same but the ECH(Expected Cost per Hour) values will change.

`echXbar` is used to calculate the ECH (Expected Cost per Hour) for one given design point.

Value

The `ecoXbar` function returns an object of class "edcc", which is a list of elements `optimum`, `cost.frame`, `FAR` and `ATS`. `optimum` is a vector with the optimum parameters and the corresponding ECH value; `cost.frame` is a dataframe with the optimum parameters and the corresponding ECH values for all given `n`(if `n` is not specified, `cost.frame` won't be returned); `FAR` indicates the false alarm rate during the in-control time, which is calculated as lambda*(average number of false alarm); `ATS` indicates the average time to signal after the occurrence of an assignable cause, calculated as h*ARL2 - tau, where tau is the expected time of occurrence of the assignable cause given it occurs between the i-th and (i+1)st samples. The `echXbar` function returns the calculated ECH value only.

References

Weicheng Zhu, Changsoon Park (2013), edcc: An R Package for the Economic Design of the Control Chart. *Journal of Statistical Software*, 52(9), 1-24. http://www.jstatsoft.org/v52/i09/

Douglas (2009). *Statistical quality control: a modern introduction*, sixth edition, 463-471.

Lorenzen and Vance (1986). The economic design of control charts: a unified approach, *Technometrics*, 28. 3-10.

See Also

ecoCusum, ecoEwma, contour, optim, update.edcc

Examples

```
# Douglas (2009). Statistical quality control: a modern introduction, sixth edition, p470.
# In control profit per hour is 110, out of control profit per hour is 10
ecoXbar(P0=110,P1=10)
# In control profit per hour is 150, out of control profit per hour
# is 50, the result is identical with the previous one, because the
#difference between P0 and P1 are the same
ecoXbar(P0=150,P1=50)
# In control cost per hour is 0, out of control cost per hour is 100.
# The result is the same with the previous one
ecoXbar(C0=0,C1=100)
# The optimum parameters are the same with the previous one,
# but Cost values are different. See 'details'
ecoXbar(C0=10,C1=110)
# Based on the global optimum above, we specify the range of the
# parameters like this
x <- ecoXbar(h=seq(0.7,0.9,by=.01),L=seq(2.8,3.3,by=.01),n=4:6,P0=110,
P1=10,nlevels=50,contour.plot=TRUE)
x
# Modify the contour plot
contour(x,nlevels=20,lty=2,col=2,call.print=FALSE)
# update the parameters
update(x,P0=NULL,P1=NULL,C0=10,C1=110)
```

---

edcc-class *Class "edcc"*

Description

A list of objects of class edcc.

Objects from the Class

Objects can be created by calling the ecoXbar, ecoCusum or ecoEwma function.

Methods

contour : creat a contour plot.

update : update the object.

print : print the object.

`update.edcc` *Update for an "edcc" class object*

Description

'update' will update and (by default) re-fit a model. It does this by extracting the call stored in the object, updating the call and (by default) evaluating that call.

Usage

```
## S3 method for class 'edcc'
update(object, ..., evaluate = TRUE)
```

Arguments

`object`	an object of "edcc" class
`...`	additional arguments to the call, or arguments with anged values.
`evaluate`	If true evaluate the new call else return the call.

Details

S3 method for update.

Value

the fitted object

Examples

```
x <- ecoXbar(P0=110,P1=10)
update(x,P0=NULL,P1=NULL,C0=10,C1=110)
```

Index