Package: spcadjust (via r-universe)

September 9, 2024

Version 1.1

Date 2015-11-20

Title Functions for Calibrating Control Charts

Author Axel Gandy <a.gandy@imperial.ac.uk> and Jan Terje Kvaloy <jan.t.kvaloy@uis.no>.

Maintainer Axel Gandy <a.gandy@imperial.ac.uk>

Depends R (>= 2.5.0)

Imports methods, stats, utils, graphics

Description Calibration of thresholds of control charts such as CUSUM charts based on past data, taking estimation error into account.

License GPL (>= 2)

Suggests knitr, parallel, testthat

VignetteBuilder knitr

Collate 'CUSUMlib.R' 'model.R' 'main.R' 'CUSUM.R' 'EWMAlib.R' 'EWMA.R' 'data.R' 'lm.R' 'logreg.R' 'shewhart.R'

RoxygenNote 5.0.1

NeedsCompilation no

Date/Publication 2016-09-29 11:37:35

Repository https://agandy.r-universe.dev

RemoteUrl https://github.com/cran/spcadjust

RemoteRef HEAD

RemoteSha de5d69322dcb921c7755154e30e5cbd659730f80

Contents

`cardiacsurgery` *Cardiac surgery data*

Description

A dataset describing the results of cardiac surgery. The data give information about the date of surgery, surgeon, Parsonnet score and outcome up to 90 days after surgery.

Usage

```
data(cardiacsurgery)
```

Format

A data frame with 5595 rows and 5 variables:

date:	date of the operation in days since the beginning of study
time:	number of days between surgery and the earlier of death and 90 days
status:	status at endpoint, 0/1 for censored/dead
Parsonnet:	Parsonnet score of the patient
surgeon:	surgeon performing the operation, numbered from 1 to 7

Source

Based on the data described in Steiner et al (2000). A subset of the data has been selected, some noise has been introduced and the follow-up was censored at 90 days.

References

Steiner SH, Cook RJ, Farewell VT, Treasure T (2000). Monitoring surgical performance using risk-adjusted cumulative sum charts. Biostat 1(4) 441-452.

`getcdfupdates` *CDF of Updates of a Control Chart*

Description

Consider running a control chart with given parameters with data coming from a given probability model. This function computes the cumulative distribution function (CDF) of the updates of the control charts as they would be computed by the method `updates`.

Usage

```
getcdfupdates(chart, P, xi)

## S4 method for signature 'SPCchart'
getcdfupdates(chart, P, xi)

## S4 method for signature 'SPCShew'
getcdfupdates(chart, P, xi)
```

Arguments

`chart`	the chart to be used.
`P`	the probability model from which data is generated.
`xi`	the parameters of the control chart.

Value

A function mapping one-dimensional numerical values into the interval [0,1], having all properties of a cumulative distribution function.

Methods (by class)

- `SPCchart`: Standard method which simply first applies getcdfresiduals from the data model.
- `SPCShew`: Computes the CDF of the updates taking into account if the chart is one-sided or two-sided.

getq *Returns a List to Compute Properties of a chart*

Description

Returns functions to compute desired properties of a given control chart.

Usage

```
getq(chart, property, params)

## S4 method for signature 'SPCCUSUM'
getq(chart, property, params)

## S4 method for signature 'SPCEWMA'
getq(chart, property, params)

## S4 method for signature 'SPCShew'
getq(chart, property, params)
```

Arguments

chart	the chart to be used.
property	the name of the property.
params	additional parameters needed for the computations.

Value

A list with the elements q, trafo, lowerconf, format.

- q(P,xi): The transformed property of interest. To improve the bootstrap a log transform is used for calARL,calhitprob and ARL, and a logit transform for hitprob. This function depends on the distribution of updates P and the chart parameters xi.
- trafo(x): The inverse of the transformation of the property used in the bootstrap. Needed to back-transform the result to the correct scale.
- lowerconf: Logical value. TRUE if a lower confidence interval should be reported, FALSE otherwise. Default is TRUE for properties calARL, calhitprob and hitprob and FALSE for ARL.
- format(res): Output summary given as a text string.

Methods (by class)

- SPCCUSUM: Implements the properties ARL, calARL, hitprob and calhitprob.
- SPCEWMA: Implements the properties ARL, calARL, hitprob and calhitprob.
- SPCShew: Implements the properties ARL, calARL, hitprob and calhitprob.

runchart *Runs a chart*

Description

Generic method for running a chart on new data using given parameters `xi`.

Usage

```
runchart(chart, newdata, xi)

## S4 method for signature 'SPCCUSUM'
runchart(chart, newdata, xi)

## S4 method for signature 'SPCEWMA'
runchart(chart, newdata, xi)

## S4 method for signature 'SPCShew'
runchart(chart, newdata, xi)
```

Arguments

`chart`	the chart to be used.
`newdata`	the new observed data.
`xi`	the parameters to be used in running the chart.

Value

The path of the chart over time.

Methods (by class)

- `SPCCUSUM`: Generic function for running CUSUM charts. Relies on `updates` being implemented for the chart.
- `SPCEWMA`: Generic function for running EWMA charts. Relies on `updates` being implemented for the chart.
- `SPCShew`: Simply computes the updates appropriate for the Shewhart chart and returns them.

SPC2sidedconfint *Computes a two-sided confidence interval for properties of a control chart.*

Description

Computes a two-sided confidence interval for properties of a control chart.

Usage

```
SPC2sidedconfint(covprob = 0.9, ...)
```

Arguments

covprob	The coverage probability of the adjustment.
...	Parameters to be passed to SPCproperty

Value

The desired confidence interval, a vector of length 2.

See Also

SPCproperty

Examples

```
# Compute 2-sided CI for the ARL of a CUSUM control chart assuming normality.
 X <-  rnorm(100) #observed data
chart <- new("SPCCUSUM",model=SPCModelNormal(Delta=1)) # CUSUM chart with normal observations
 SPC2sidedconfint(data=X,nrep=100,covprob=0.95,
            property="ARL",chart=chart,params=list(threshold=4))
```

SPCchart-class *Virtual Base Class for Control Charts*

Description

Virtual S4 base class for Control Charts.

Slots

model The data model to be used in the chart. Must be of type SPCDataModel.

SPCCUSUM-class *CUSUM Charts*

Description

Class extending SPCChart with a basic CUSUM charts implementation.

Details

The only slot this class contains is the data model. This data model should already incorporate the negative mean for in-control updates that is typical for CUSUM charts.

Let $U_t, t = 1, 2, \ldots$ be the updates from the data model. Then the CUSUM chart is given by $S_0 = 0$ and

$$S_t = max(S_{t-1} + U_t, 0)$$

Examples

```
X <-  rnorm(1000)
chart <- new("SPCCUSUM",model=SPCModelNormal(Delta=1))
## Not run:
SPCproperty(data=X,nrep=10,chart=chart,
            property="calARL",params=list(target=100))
SPCproperty(data=X,nrep=10,chart=chart,
            property="calhitprob",params=list(target=0.05,nsteps=1e3))
SPCproperty(data=X,nrep=10,chart=chart,
            property="ARL",params=list(threshold=3))

## End(Not run)
SPCproperty(data=X,nrep=10,chart=chart,
            property="hitprob",params=list(threshold=3,nsteps=1e3))
#increase the number of repetitions nrep for real applications.
```

SPCDataModel-class *Data Model for SPC charts*

Description

This is the basic structure for converting observations (data) into updates for control charts. Classes of this type also have the ability to generate new data sets (resampling).

Details

Every element of this class has to consist of a list of the following functions: updates, getcdfupdates, Pofdata, resample, xiofP, which have to be of a specific form. The arguments generally have the following meaning: `xi` denotes the parameter vector needed to create updates for running the chart from observed data, `data` is observed data, `P` is a data model.

- `updates(xi,data)`: Returns updates for the chart using the parameter `xi` and the observed data `data`.
- `Pofdata(data)`: Estimates a probability model from the data.
- xiofP(P): Computes the parameter `xi` needed to compute updates from an (estimated) probability model `P`.
- `resample(P)`: Generates a new data set from the probability model `P`.
- `getcdfupdates(P,xi,cadlag=TRUE)`: Returns the cumulative distribution function (CDF) of updates of data generated from the probability model `P` and computed using the parameter `xi`. The CDF has to be a function of one argument that also accepts vectors. If cadlag is TRUE then the CDF is right-continuous (i.e. $F(x) = P(X \leq x)$). If cadlag is FALSE then the CDF is left-continuous (i.e. $F(x) = P(X < x)$).

See Also

SPCModelNormal, SPCModelNonpar, SPCModelNonparCenterScale

SPCEWMA-class *EWMA Charts*

Description

Class extending SPCChart with a basic EWMA charts implementation.

Details

Let $Y_t, t = 1, 2, \ldots$ be the updates from the data model. Then the EWMA chart is given by $Q_0 = 0$ and

$$Q_t = lambda Y_t + (1 - lambda) Q_{t-1}$$

Slots

`model` The data model. The data model should center the in-control updates such that they have mean 0.

`lambda` The smoothing constant, $0 < lambda <= 1$.

Examples

```
X <-  rnorm(1000)
chart <- new("SPCEWMA",model=SPCModelNormal(Delta=0),lambda=0.8)
## Not run:
SPCproperty(data=X,nrep=10,chart=chart,
            property="calARL",params=list(target=100))
SPCproperty(data=X,nrep=10,chart=chart,
            property="calhitprob",params=list(target=0.05,nsteps=1e3))

## End(Not run)
SPCproperty(data=X,nrep=10,chart=chart,
            property="ARL",params=list(threshold=3))
SPCproperty(data=X,nrep=10,chart=chart,
            property="hitprob",params=list(threshold=3,nsteps=1e3))
#increase the number of repetitions nrep for real applications.
```

SPCModellm *Data model based on a linear model*

Description

The parameters needed for running the chart are the fitted linear model. Resampled data sets are created by resampling cases with replacement (i.e. keeping observations together).

Usage

```
SPCModellm(formula, Delta = 0)
```

Arguments

formula	the linear model specified as a formula.
Delta	Object of class "numeric".

Examples

```
n <- 1000
Xlinreg <- data.frame(x1= rbinom(n,1,0.4), x2= runif(n,0,1), x3= rnorm(n))
Xlinreg$y <- 2 + Xlinreg$x1 + Xlinreg$x2 + Xlinreg$x3 + rnorm(n)
## Not run:
chartlinregCUSUM <- new("SPCCUSUM", model=SPCModellm(Delta=1,formula="y~x1+x2+x3"))
SPCproperty(data=Xlinreg,nrep=10,property="calARL",
            chart=chartlinregCUSUM,params=list(target=100))
#increase nrep in real applications.
#' chartlinregCUSUM2 <- new("SPCCUSUM",model=SPCModellm(Delta=1,formula="y~x1"))
SPCproperty(data=Xlinreg,nrep=10,property="calARL",
            chart=chartlinregCUSUM2,params=list(target=100))
#increase nrep in real applications.
```

```
chartlinregEWMA <- new("SPCEWMA", model=SPCModellm(Delta=0,formula="y~x1+x2+x3"),lambda=0.8)
SPCproperty(data=Xlinreg,nrep=10,property="calARL",
            chart=chartlinregEWMA,params=list(target=100))
#increase nrep in real applications.

chartlinregEWMA2 <- new("SPCEWMA",model=SPCModellm(Delta=0,formula="y~x1"),lambda=0.8)
SPCproperty(data=Xlinreg,nrep=10,property="calARL",
            chart=chartlinregEWMA2,params=list(target=100))

## End(Not run)
#increase nrep in real applications.
```

`SPCModellogregLikRatio` *Data Model for Binary Responses using a logarithmic model and likelihood ratio updates.*

Description

Data Model for Binary Responses using a logarithmic model and likelihood ratio updates.

Usage

```
SPCModellogregLikRatio(formula, Delta = 1)
```

Arguments

`formula`	The formula of the model.
`Delta`	This value will be added to the log odds ratio in the out-of-control model in the likelihood ratio between the out-of-control and the in-control model constituting the updates.

Examples

```
n <- 1000
Xlogreg <- data.frame(x1=rbinom(n,1,0.4), x2=runif(n,0,1), x3=rnorm(n))
xbeta <- -1+Xlogreg$x1*100+Xlogreg$x2+Xlogreg$x3
Xlogreg$y <- rbinom(n,1,exp(xbeta)/(1+exp(xbeta)))
chartlogreg <- new("SPCCUSUM",
                   model=SPCModellogregLikRatio(Delta= 1, formula="y~x1+x2+x3"))
SPCproperty(data=Xlogreg,nrep=10,property="calARL",
            chart=chartlogreg,params=list(target=100))
#increase nrep for real applications.
```

SPCModellogregOE — *Data Model for Binary Responses using a Logarithmic Model and observed minus expected updates.*

Description

Data Model for Binary Responses using a Logarithmic Model and observed minus expected updates.

Usage

```
SPCModellogregOE(formula, Delta = 0)
```

Arguments

formula	The formula of the model.
Delta	Half of this value will be subtracted for every update.

Examples

```
n <- 1000
Xlogreg <- data.frame(x1=rbinom(n,1,0.4), x2=runif(n,0,1), x3=rnorm(n))
xbeta <- -1+Xlogreg$x1*100+Xlogreg$x2+Xlogreg$x3
Xlogreg$y <- rbinom(n,1,exp(xbeta)/(1+exp(xbeta)))
chartlogreg <- new("SPCEWMA",
                   model=SPCModellogregOE(Delta= 0, formula="y~x1+x2+x3"), lambda=0.8)
SPCproperty(data=Xlogreg,nrep=10,property="calARL",
            chart=chartlogreg,params=list(target=100))
#increase nrep for real applications.
```

SPCModelNonpar — *Generic Model for Nonparametric Resampling*

Description

Generic model that allows nonparametric resampling (with replacement) of the data. The transformation of data into updates needs to be defined via the arguments.

Usage

```
SPCModelNonpar(updates, xiofP)
```

Arguments

updates	function that computes updates.
xiofP	function that computes xi given P.

Details

The parameters to the functions being returned have the following meaning.

- data: a numeric vector or a matrix where the rows contain the observations.
- xi: depends on the parameter xiofP.
- P: The `data` with no modifications (thus either a numeric vector or a matrix).

The main operations are defined as follows:

- resample(P): generates a new data set of the same size by either resampling the values (if the data is a vector) or by resampling the rows of the data matrix (both use resampling with replacement).

Value

An object of class SPCDataModel.

`SPCModelNonparCenterScale` *Nonparametric Resampling with Centering and Scaling*

Description

Nonparametric resampling of univariate observations. Updates are centered and scaled transformations of the data (with a constant potentially being subtracted).

Usage

```
SPCModelNonparCenterScale(Delta = 0)
```

Arguments

`Delta`	how much to subtract before scaling.

Details

Calls `SPCModelNonpar` to generate the data object, so it only needs to specify the meaning of `xi`, the parameter needed to compute updates and the definition of the updates.

- xi: a list with two elements:
  - mu: the mean.
  - sd: the standard deviation.
- updates(xi,data): returns the centered and scale version of the data from which $Delta/2$ has been subtracted, i.e.

$$\frac{data - mu - Delta/2}{sd}$$

Value

An object of class SPCDataModel.

An object of class SPCDataModel.

See Also

SPCModelNonpar

Examples

```
X <- rnorm(1000)

#CUSUM chart
model <- SPCModelNonparCenterScale(1)
chart <- new("SPCCUSUM",model=model)
SPCproperty(data=X,nrep=10,property="calARL",
            chart=chart,params=list(target=100))

#Shewhart chart
model <- SPCModelNonparCenterScale(0)
chart <- new("SPCCUSUM",model=model)
SPCproperty(data=X,nrep=10,property="calARL",
            chart=chart,params=list(target=100))
```

SPCModelNormal *Parametric Model for Normally Distributed Data with Centering and Scaling*

Description

Returns a data model for univariate observations using normality assumptions with updates that center and scale the observations and potentially subtract half of a constant `Delta`. Subtracting $Delta/2$ is useful for CUSUM charts.

Usage

```
SPCModelNormal(Delta = 0)
```

Arguments

`Delta`	Half of this constant is subtracted for updates (before centering and scaling).

Details

The parameters to the function have the following meaning.

- data: a numeric vector.
- xi: a list with two elements:

  - mu: the mean.
  - sd: the standard deviation.
- P: a list with three elements:
  - mu: the mean.
  - sd: the standard deviation.
  - m: the number of data points to resample.

The main operations are defined as follows:

- updates(xi,data): returns the centered and scale version of the data from which $Delta/2$ has been subtracted, i.e.
$$\frac{data - mu - Delta/2}{sd}$$
.
- resample(P): resamples `m` new data points from a normal distribution if mean `mu` and standard deviation `sd`.

Value

An object of class `SPCDataModel`.

`SPCproperty` *Computes bootstrap adjusted properties for control charts*

Description

Computes bootstrap adjusted properties for control charts.

Usage

```
SPCproperty(data, nrep = 500, chart, property, params, covprob = 0.9,
  quiet = FALSE, reportdistr = FALSE, parallel = 1)
```

Arguments

`data`	The observed data.
`nrep`	The number of bootstrap repetitions. Default 500.
`chart`	The chart to be used.
`property`	The property to be computed. A string. Must be implemented by the chart.
`params`	Additional parameters for computing the property.
`covprob`	The coverage probability of the adjustment. Default 0.9.
`quiet`	Logical value indicating if progress bar should be suppressed. Default FALSE.
`reportdistr`	Logical value indicating if the ecdf of the bootstrap distribution should be plotted. Default FALSE.
`parallel`	Number of cores to use for parallel computations (using mclapply from the package parallel). Defaults to 1. If set to `Inf` then the number of cores is automatically detected and all but one are used. Has to be set to 1 under Windows.

Value

An object of type SPCpropertyres.

See Also

SPC2sidedconfint

Examples

```
 # calibrate CUSUM chart to an in-control ARL of 100.
 # run with a larger number of replications in real examples!
 X <-  rnorm(100) #observed data
 chart <- new("SPCCUSUM",model=SPCModelNormal(Delta=1)) # CUSUM chart with normal observations
SPCproperty(data=X,nrep=15,chart=chart,property="calARL", params=list(target=100))
```

SPCpropertyres-class *Results of SPCproperty.*

Description

Results of SPCproperty.

Usage

```
## S4 method for signature 'SPCpropertyres'
show(object)
```

Arguments

object	the result to be shown.

Methods (by generic)

- show: Prints the object nicely.

Slots

nrep number of repetitions used in the simulation.

chart the chart used.

property the property of interest, ARL, calARL, calhitprob or hitprob.

covprob the probability of the guaranteed conditional performance.

res the guaranteed conditional performance.

raw the unadjusted result.

params additional parameters used for computing this property.

restext a readable version of the result.

SPCShew-class *Shewhart charts.*

Description

Shewhart charts.

Slots

`twosided` TRUE if a two-sided chart should be used. Default FALSE.

Examples

```
X<-rnorm(100);
##calibrate to ARL 100
chartShew <- new("SPCShew",model=SPCModelNormal(),twosided=TRUE)
## Not run:
SPCproperty(data=X,nrep=500,
            property="calARL",chart=chartShew,params=list(target=100),
            covprob=c(0.7,0.9))

chartShewOneSided <- new("SPCShew",model=SPCModelNormal(),twosided=FALSE)
SPCproperty(data=X,nrep=500,
            property="calARL",chart=chartShewOneSided,
            params=list(target=100),covprob=c(0.7,0.9))

##calibrate to a hitting probability of 0.01 in 100 steps
SPCproperty(data=X,nrep=500,
            property="calhitprob",
            chart=chartShew,params=list(target=0.01,nsteps=100))
SPCproperty(data=X,nrep=500,
        property="calhitprob",chart=chartShewOneSided,params=list(target=0.01,nsteps=100))

## work out  for ARL for a fixed threshold of 4
SPCproperty(data=X,nrep=500,
            property="ARL",chart=chartShew,params=list(threshold=4))
SPCproperty(data=X,nrep=500,
            property="ARL",chart=chartShewOneSided,
            params=list(threshold=4))

SPCproperty(data=X,nrep=500,
            property="hitprob",chart=chartShew,params=list(nsteps=100,threshold=4))

SPCproperty(data=X,nrep=500,
         property="hitprob",chart=chartShewOneSided,params=list(nsteps=100,threshold=4))

## End(Not run)

X<-rnorm(100)
chartShew <- new("SPCShew",model=SPCModelNormal())
```

```
## Not run:
SPCproperty(data=X,nrep=500,
            property="calARL", chart=chartShew,
            params=list(target=1000))
SPCproperty(data=X,nrep=500,
            property="calhitprob",chart=chartShew,
            params=list(target=0.01,nsteps=100))
SPCproperty(data=X,nrep=10,chart=chartShew,
            property="ARL",params=list(threshold=3))
SPCproperty(data=X,nrep=500,
            property="hitprob",
            chart=chartShew,params=list(nsteps=100,threshold=4))

## End(Not run)
```

`updates` *Updates of a Control Chart*

Description

Computes updates of a control chart using the given parameters and the given data.

Usage

```
updates(chart, xi, data)

## S4 method for signature 'SPCchart'
updates(chart, xi, data)

## S4 method for signature 'SPCShew'
updates(chart, xi, data)
```

Arguments

`chart`	the control chart.
`xi`	the parameters used for running the chart.
`data`	the observed data.

Value

A vector of the same length as data.

Methods (by class)

- `SPCchart`: Standard method which simply first applies getresiduals from the data model.
- `SPCShew`: Computes the updates taking into account if the chart is one-sided or two-sided.

`xiofdata` *Estimate Chart Parameters*

Description

Estimates the parameters used to run a control chart from given data.

Usage

```
xiofdata(chart, data)

## S4 method for signature 'SPCchart'
xiofdata(chart, data)
```

Arguments

`chart`	the control chart to be used.
`data`	the data to be used.

Value

The parameter values for running the control chart `chart`.

Methods (by class)

- `SPCchart`: Standard method which simply first applies PofData to get a model and then xiofP to get the parameters.

Index