Package 'qcc'

May 9, 2026

Version 2.7

Date 2017-07-09

Title Quality Control Charts

Description Shewhart quality control charts for continuous, attribute and count data. Cusum and EWMA charts. Operating characteristic curves. Process capability analysis. Pareto chart and cause-and-effect chart. Multivariate control charts.

Depends R (>= 3.0)

Imports MASS, utils, graphics, grDevices

Suggests knitr (>= 1.12), rmarkdown (>= 0.9)

License GPL (>= 2)

VignetteBuilder knitr

URL https://github.com/luca-scr/qcc

Repository CRAN

ByteCompile true

LazyLoad yes

NeedsCompilation no

Author Luca Scrucca [aut, cre],
Greg Snow [ctb],
Peter Bloomfield [ctb]

Maintainer Luca Scrucca <luca.scrucca@unipg.it>

Date/Publication 2017-07-11 05:51:24 UTC

Contents

`qcc-package` *Quality Control Charts*

Description

Shewhart quality control charts for continuous, attribute and count data. Cusum and EWMA charts. Operating characteristic curves. Process capability analysis. Pareto chart and cause-and-effect chart. Multivariate control charts.

Details

See vignette and documentation accompanying the package.

Author(s)

Luca Scrucca

References

Scrucca, L. (2004). qcc: an R package for quality control charting and statistical process control. *R News* 4/1, 11-17.

See Also

`qcc`, `mqcc`, `cusum`, `ewma`, `oc.curves`, `process.capability`, `pareto.chart`, `cause.and.effect`.

`boiler` *Boiler temperature data*

Description

Temperature readings from the eight configured burners on a boiler.

Usage

```
data(boiler)
```

Format

A data frame with 25 observations on the following 8 variables:

t1 temperature reading 1

t2 temperature reading 2

t3 temperature reading 3

t4 temperature reading 4

t5 temperature reading 5

t6 temperature reading 6

t7 temperature reading 7

t8 temperature reading 8

References

Mason, R.L. and Young, J.C. (2002) *Multivariate Statistical Process Control with Industrial Applications*, SIAM, p. 86.

Examples

```
data(boiler)
summary(boiler)
boxplot(boiler)
```

`cause.and.effect` *Cause and effect diagram*

Description

Draw a basic cause and effect diagram.

Usage

```
cause.and.effect(cause, effect, title = "Cause-and-Effect diagram",
                 cex = c(1, 0.9, 1), font = c(1, 3, 2))
```

Arguments

`cause`	a list of causes and branches providing descriptive labels (see the example below).
`effect`	a string label or the effect.
`title`	a string specifying the main title to appear on the plot.
`cex`	a vector of values for the graphical character expansion. The values refer, in order, to branches, causes and effect.
`font`	a vector of values for the font to use. The values refer, in order, to branches, causes and effect.

Author(s)

Luca Scrucca

References

Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

Examples

```
cause.and.effect(cause=list(Measurements=c("Micrometers", "Microscopes", "Inspectors"),
                            Materials=c("Alloys", "Lubricants", "Suppliers"),
                          Personnel=c("Shifts", "Supervisors", "Training", "Operators"),
                            Environment=c("Condensation", "Moisture"),
                            Methods=c("Brake", "Engager", "Angle"),
                            Machines=c("Speed", "Lathes", "Bits", "Sockets")),
                 effect="Surface Flaws")
```

circuit *Circuit boards data*

Description

Number of nonconformities observed in 26 successive samples of 100 printed circuit boards. Sample 6 and 20 are outside the control limits. Sample 6 was examined by a new inspector and he did not recognize several type of nonconformities that could have been present. Furthermore, the unusually large number of nonconformities in sample 20 resulted from a temperature control problem in the wave soldering machine, which was subsequently repaired. The last 20 samples are further samples collected on inspection units (each formed by 100 boards).

Usage

```
data(circuit)
```

Format

A data frame with 46 observations on the following 3 variables.

x number of defectives in 100 printed circuit boards (inspection unit)

size sample size

trial trial sample indicator (TRUE/FALSE)

References

Montgomery, D.C. (1991) *Introduction to Statistical Quality Control*, 2nd ed, New York, John Wiley & Sons, pp. 173–175

Examples

```
data(circuit)
attach(circuit)
summary(circuit)
boxplot(x ~ trial)
plot(x, type="b")
detach(circuit)
```

cusum *Cusum chart*

Description

Create an object of class 'cusum.qcc' to compute a Cusum chart for statistical quality control.

Usage

```
cusum(data, sizes, center, std.dev, head.start = 0,
      decision.interval = 5, se.shift = 1, data.name, labels,
      newdata, newsizes, newlabels, plot = TRUE, ...)

## S3 method for class 'cusum.qcc'
print(x, ...)

## S3 method for class 'cusum.qcc'
summary(object, digits = getOption("digits"), ...)

## S3 method for class 'cusum.qcc'
plot(x, add.stats = TRUE, chart.all = TRUE,
     label.bounds = c("LDB", "UDB"), title, xlab, ylab, ylim,
     axes.las = 0, digits =  getOption("digits"),
     restore.par = TRUE, ...)
```

Arguments

data	a data frame, a matrix or a vector containing observed data for the variable to chart. Each row of a data frame or a matrix, and each value of a vector, refers to a sample or "rationale group".
sizes	a value or a vector of values specifying the sample sizes associated with each group. If not provided the sample sizes are obtained counting the non-NA elements of each row of a data frame or a matrix; sample sizes are set all equal to one if data is a vector.
center	a value specifying the center of group statistics or the "target" value of the process.
std.dev	a value or an available method specifying the within-group standard deviation(s) of the process. Several methods are available for estimating the standard deviation. See sd.xbar and sd.xbar.one for, respectively, the grouped data case and the individual observations case.
head.start	The initializing value for the above-target and below-target cumulative sums, measured in standard errors of the summary statistics. Use zero for the traditional Cusum chart, or a positive value less than the decision.interval for a Fast Initial Response.

`decision.interval`
: A numeric value specifying the number of standard errors of the summary statistics at which the cumulative sum is out of control.

`se.shift`
: The amount of shift to detect in the process, measured in standard errors of the summary statistics.

`data.name`
: a string specifying the name of the variable which appears on the plots. If not provided is taken from the object given as data.

`labels`
: a character vector of labels for each group.

`newdata`
: a data frame, matrix or vector, as for the `data` argument, providing further data to plot but not included in the computations.

`newsizes`
: a vector as for the `sizes` argument providing further data sizes to plot but not included in the computations.

`newlabels`
: a character vector of labels for each new group defined in the argument `newdata`.

`plot`
: logical. If `TRUE` a Cusum chart is plotted.

`add.stats`
: a logical value indicating whether statistics and other information should be printed at the bottom of the chart.

`chart.all`
: a logical value indicating whether both statistics for `data` and for `newdata` (if given) should be plotted.

`label.bounds`
: a character vector specifying the labels for the the decision interval boundaries.

`title`
: a string giving the label for the main title.

`xlab`
: a string giving the label for the x-axis.

`ylab`
: a string giving the label for the y-axis.

`ylim`
: a numeric vector specifying the limits for the y-axis.

`axes.las`
: numeric in {0,1,2,3} specifying the style of axis labels. See `help(par)`.

`digits`
: the number of significant digits to use.

`restore.par`
: a logical value indicating whether the previous `par` settings must be restored. If you need to add points, lines, etc. to a control chart set this to `FALSE`.

`object`
: an object of class 'cusum.qcc'.

`x`
: an object of class 'cusum.qcc'.

`...`
: additional arguments to be passed to the generic function.

Details

Cusum charts display how the group summary statistics deviate above or below the process center or target value, relative to the standard errors of the summary statistics. Useful to detect small and permanent variation on the mean of the process.

Value

Returns an object of class 'cusum.qcc'.

Author(s)

Luca Scrucca

References

Mason, R.L. and Young, J.C. (2002) *Multivariate Statistical Process Control with Industrial Applications*, SIAM.
Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Ryan, T. P. (2000), *Statistical Methods for Quality Improvement*, 2nd ed. New York: John Wiley & Sons, Inc.
Scrucca, L. (2004). qcc: an R package for quality control charting and statistical process control. *R News* 4/1, 11-17.
Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

See Also

qcc, ewma

Examples

```
##
## Grouped-data
##
data(pistonrings)
attach(pistonrings)
diameter <- qcc.groups(diameter, sample)

q <- cusum(diameter[1:25,], decision.interval = 4, se.shift = 1)
summary(q)

q <- cusum(diameter[1:25,], newdata=diameter[26:40,])
summary(q)
plot(q, chart.all=FALSE)

detach(pistonrings)
```

---

dyedcloth *Dyed cloth data*

Description

In a textile finishing plant, dyed cloth is inspected for the occurrence of defects per 50 square meters. The data on ten rolls of cloth are presented.

Usage

```
data(dyedcloth)
```

Format

A data frame with 10 observations on the following 2 variables.

x number of nonconformities per 50 square meters (inspection units)

size number of inspection units in roll (variable sample size)

References

Montgomery, D.C. (1991) *Introduction to Statistical Quality Control*, 2nd ed, New York, John Wiley & Sons, pp. 183–184

Examples

```
data(dyedcloth)
attach(dyedcloth)
summary(dyedcloth)
plot(x/size, type="b")
detach(dyedcloth)
```

---

ellipseChart *Multivariate Quality Control Charts*

Description

Plot an ellipse chart for a bivariate quality control data.

Usage

```
ellipseChart(object, chart.all = TRUE, show.id = FALSE, ngrid = 50,
             confidence.level, correct.multiple = TRUE,
             title, xlim, ylim, xlab, ylab,
             restore.par = TRUE, ...)
```

Arguments

`object`	an object of class 'mqcc'.
`chart.all`	a logical value indicating whether both statistics for `data` and for `newdata` (if given) should be plotted.
`show.id`	a logical value indicating whether to plot point labels (`TRUE`) or symbols (`FALSE`) for group means.
`ngrid`	a value for the size of the grid over which the ellipse is evaluated.
`confidence.level`	a numeric value between 0 and 1 specifying the confidence level of the computed probability limits.
`correct.multiple`	a logical value indicating whether to correct or not for multiple comparisons.

`title`	a string giving the label for the main title.
`xlim`	a numeric vector specifying the limits for the x-axis.
`ylim`	a numeric vector specifying the limits for the y-axis.
`xlab`	a string giving the label for the x-axis.
`ylab`	a string giving the label for the y-axis.
`restore.par`	a logical value indicating whether the previous `par` settings must be restored. If you need to add points, lines, etc. to a control chart set this to `FALSE`.
`...`	additional arguments to be passed to the generic `points` function.

Author(s)

Luca Scrucca

References

Mason, R.L. and Young, J.C. (2002) *Multivariate Statistical Process Control with Industrial Applications*, SIAM.
Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Ryan, T. P. (2000), *Statistical Methods for Quality Improvement*, 2nd ed. New York: John Wiley & Sons, Inc.

See Also

`mqcc`, `stats.T2`, `stats.T2.single`

Examples

```
# See examples in help(mqcc)
```

---

ewma *EWMA chart*

Description

Create an object of class 'ewma.qcc' to compute and draw an Exponential Weighted Moving Average (EWMA) chart for statistical quality control.

Usage

```
ewma(data, sizes, center, std.dev, lambda = 0.2, nsigmas = 3,
     data.name, labels, newdata, newsizes, newlabels,
     plot = TRUE, ...)

## S3 method for class 'ewma.qcc'
print(x, ...)
```

```
## S3 method for class 'ewma.qcc'
summary(object, digits = getOption("digits"), ...)

## S3 method for class 'ewma.qcc'
plot(x, add.stats = TRUE, chart.all = TRUE,
     label.limits = c("LCL", "UCL"), title, xlab, ylab, ylim,
     axes.las = 0, digits = getOption("digits"),
     restore.par = TRUE, ...)
```

Arguments

data	a data frame, a matrix or a vector containing observed data for the variable to chart. Each row of a data frame or a matrix, and each value of a vector, refers to a sample or "rationale group".
sizes	a value or a vector of values specifying the sample sizes associated with each group. If not provided the sample sizes are obtained counting the non-NA elements of each row of a data frame or a matrix; sample sizes are set all equal to one if data is a vector.
center	a value specifying the center of group statistics or target.
std.dev	a value or an available method specifying the within-group standard deviation(s) of the process. Several methods are available for estimating the standard deviation. See sd.xbar and sd.xbar.one for, respectively, the grouped data case and the individual observations case.
lambda	the smoothing parameter $0 \leq \lambda \leq 1$
nsigmas	a numeric value specifying the number of sigmas to use for computing control limits.
data.name	a string specifying the name of the variable which appears on the plots. If not provided is taken from the object given as data.
labels	a character vector of labels for each group.
newdata	a data frame, matrix or vector, as for the data argument, providing further data to plot but not included in the computations.
newsizes	a vector as for the sizes argument providing further data sizes to plot but not included in the computations.
newlabels	a character vector of labels for each new group defined in the argument newdata.
plot	logical. If TRUE an EWMA chart is plotted.
add.stats	a logical value indicating whether statistics and other information should be printed at the bottom of the chart.
chart.all	a logical value indicating whether both statistics for data and for newdata (if given) should be plotted.
label.limits	a character vector specifying the labels for control limits.
title	a string giving the label for the main title.
xlab	a string giving the label for the x-axis.

ylab	a string giving the label for the y-axis.
ylim	a numeric vector specifying the limits for the y-axis.
axes.las	numeric in {0,1,2,3} specifying the style of axis labels. See help(par).
digits	the number of significant digits to use.
restore.par	a logical value indicating whether the previous par settings must be restored. If you need to add points, lines, etc. to a control chart set this to FALSE.
object	an object of class 'ewma.qcc'.
x	an object of class 'ewma.qcc'.
...	additional arguments to be passed to the generic function.

Details

EWMA chart smooths a series of data based on a moving average with weights which decay exponentially. Useful to detect small and permanent variation on the mean of the process.

Value

Returns an object of class 'ewma.qcc'.

Author(s)

Luca Scrucca

References

Mason, R.L. and Young, J.C. (2002) *Multivariate Statistical Process Control with Industrial Applications*, SIAM.
Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Ryan, T. P. (2000), *Statistical Methods for Quality Improvement*, 2nd ed. New York: John Wiley & Sons, Inc.
Scrucca, L. (2004). qcc: an R package for quality control charting and statistical process control. *R News* 4/1, 11-17.
Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

See Also

qcc, ewmaSmooth, cusum

Examples

```
##
## Grouped-data
##
data(pistonrings)
attach(pistonrings)
diameter <- qcc.groups(diameter, sample)
```

```
q <- ewma(diameter[1:25,], lambda=0.2, nsigmas=3)
summary(q)

q <- ewma(diameter[1:25,], lambda=0.2, nsigmas=2.7, newdata=diameter[26:40,], plot = FALSE)
summary(q)
plot(q)

detach(pistonrings)

##
## Individual observations: viscosity data (Montgomery, pag. 242)
##
x <- c(33.75, 33.05, 34, 33.81, 33.46, 34.02, 33.68,
33.27, 33.49, 33.20, 33.62, 33.00, 33.54, 33.12, 33.84)
q <-  ewma(x, lambda=0.2, nsigmas=2.7)
summary(q)
```

---

ewmaSmooth *EWMA smoothing function*

Description

Compute Exponential Weighted Moving Average.

Usage

```
ewmaSmooth(x, y, lambda = 0.2, start, ...)
```

Arguments

x	a vector of x-values.
y	a vector of y-values.
lambda	the smoothing parameter.
start	the starting value.
...	additional arguments (currently not used).

Details

EWMA function smooths a series of data based on a moving average with weights which decay exponentially.

For each y_t value the smoothed value is computed as

$$z_t = \lambda y_t + (1 - \lambda) z_{t-1}$$

where $0 \leq \lambda \leq 1$ is the parameter which controls the weights applied.

Value

Returns a list with elements:

x	ordered x-values
y	smoothed y-values
lambda	the smoothing parameter
start	the starting value

Author(s)

Luca Scrucca

References

Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

See Also

qcc, cusum

Examples

```
x <- 1:50
y <- rnorm(50, sin(x/5), 0.5)
plot(x,y)
lines(ewmaSmooth(x,y,lambda=0.1), col="red")
```

---

mqcc *Multivariate Quality Control Charts*

Description

Create an object of class 'mqcc' to perform multivariate statistical quality control.

Usage

```
mqcc(data, type = c("T2", "T2.single"), center, cov,
     limits = TRUE, pred.limits = FALSE,
     data.name, labels, newdata, newlabels,
     confidence.level = (1 - 0.0027)^p, rules = shewhart.rules,
     plot = TRUE, ...)

## S3 method for class 'mqcc'
print(x, ...)
```

```
## S3 method for class 'mqcc'
summary(object, digits = getOption("digits"), ...)

## S3 method for class 'mqcc'
plot(x, add.stats = TRUE, chart.all = TRUE,
     label.limits = c("LCL", "UCL"), label.pred.limits = c("LPL", "UPL"),
     title, xlab, ylab, ylim, axes.las = 0,
     digits = getOption("digits"), restore.par = TRUE, ...)
```

Arguments

`data`	For subgrouped data, a list with a data frame or a matrix for each variable to monitor. Each row of the data frame or matrix refers to a sample or "rationale" group. For individual observations, where each sample has a single observation, users can provide a list with a data frame or a matrix having a single column, or a data frame or a matrix where each rows refer to samples and columns to variables. See examples.
`type`	a character string specifying the type of chart:

	Chart description
`"T2"`	Hotelling T^2 chart for subgrouped data
`"T2.single"`	Hotelling T^2 chart for individual observations

`center`	a vector of values to use for center of input variables.
`cov`	a matrix of values to use for the covariance matrix of input variables.
`limits`	a logical indicating if control limits (Phase I) must be computed (by default using `limits.T2` or `limits.T2.single`) and plotted, or a two-values vector specifying control limits.
`pred.limits`	a logical indicating if prediction limits (Phase II) must be computed (by default using `limits.T2` or `limits.T2.single`) and plotted, or a two-values vector specifying prediction limits.
`data.name`	a string specifying the name of the variable which appears on the plots. If not provided is taken from the object given as data.
`labels`	a character vector of labels for each group.
`newdata`	a data frame, matrix or vector, as for the `data` argument, providing further data to plot but not included in the computations.
`newlabels`	a character vector of labels for each new group defined in the argument `newdata`.
`confidence.level`	a numeric value between 0 and 1 specifying the confidence level of the computed probability limits. By default is set at $(1 - 0.0027)^p$ where p is the number of variables, and 0.0027 is the probability of Type I error for a single Shewhart chart at the usual 3-sigma control level.
`rules`	a function of rules to apply to the chart. By default, the `shewhart.rules` function is used.

plot	logical. If TRUE a quality chart is plotted.
add.stats	a logical value indicating whether statistics and other information should be printed at the bottom of the chart.
chart.all	a logical value indicating whether both statistics for data and for newdata (if given) should be plotted.
label.limits	a character vector specifying the labels for control limits (Phase I).
label.pred.limits	a character vector specifying the labels for prediction control limits (Phase II).
title	a string giving the label for the main title.
xlab	a string giving the label for the x-axis.
ylab	a string giving the label for the y-axis.
ylim	a numeric vector specifying the limits for the y-axis.
axes.las	numeric in {0,1,2,3} specifying the style of axis labels. See help(par).
digits	the number of significant digits to use when add.stats = TRUE.
restore.par	a logical value indicating whether the previous par settings must be restored. If you need to add points, lines, etc. to a control chart set this to FALSE.
object	an object of class 'mqcc'.
x	an object of class 'mqcc'.
...	additional arguments to be passed to the generic function.

Value

Returns an object of class 'mqcc'.

Author(s)

Luca Scrucca

References

Mason, R.L. and Young, J.C. (2002) *Multivariate Statistical Process Control with Industrial Applications*, SIAM.
Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Ryan, T. P. (2000), *Statistical Methods for Quality Improvement*, 2nd ed. New York: John Wiley & Sons, Inc.
Scrucca, L. (2004). qcc: an R package for quality control charting and statistical process control. *R News* 4/1, 11-17.
Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

See Also

stats.T2, stats.T2.single, limits.T2, limits.T2.single, ellipseChart, qcc

Examples

```
##
##  Subgrouped data
##

# Ryan (2000, Table 9.2) data with p = 2 variables, m = 20 samples, n = 4 sample size:
X1 <- matrix(c(72, 56, 55, 44, 97, 83, 47, 88, 57, 26, 46,
49, 71, 71, 67, 55, 49, 72, 61, 35, 84, 87, 73, 80, 26, 89, 66,
50, 47, 39, 27, 62, 63, 58, 69, 63, 51, 80, 74, 38, 79, 33, 22,
54, 48, 91, 53, 84, 41, 52, 63, 78, 82, 69, 70, 72, 55, 61, 62,
41, 49, 42, 60, 74, 58, 62, 58, 69, 46, 48, 34, 87, 55, 70, 94,
49, 76, 59, 57, 46), ncol = 4)
X2 <- matrix(c(23, 14, 13, 9, 36, 30, 12, 31, 14, 7, 10,
11, 22, 21, 18, 15, 13, 22, 19, 10, 30, 31, 22, 28, 10, 35, 18,
11, 10, 11, 8, 20, 16, 19, 19, 16, 14, 28, 20, 11, 28, 8, 6,
15, 14, 36, 14, 30, 8, 35, 19, 27, 31, 17, 18, 20, 16, 18, 16,
13, 10, 9, 16, 25, 15, 18, 16, 19, 10, 30, 9, 31, 15, 20, 35,
12, 26, 17, 14, 16), ncol = 4)
X <- list(X1 = X1, X2 = X2)

q <- mqcc(X, type = "T2")
summary(q)
ellipseChart(q)
ellipseChart(q, show.id = TRUE)

q <- mqcc(X, type = "T2", pred.limits = TRUE)

# Ryan (2000) discussed Xbar-charts for single variables computed adjusting the
# confidence level of the T^2 chart:
q1 <- qcc(X1, type = "xbar", confidence.level = q$confidence.level^(1/2))
summary(q1)
q2 <- qcc(X2, type = "xbar", confidence.level = q$confidence.level^(1/2))
summary(q2)


require(MASS)
# generate new "in control" data
Xnew <- list(X1 = matrix(NA, 10, 4), X2 =  matrix(NA, 10, 4))
for(i in 1:4)
   { x <- mvrnorm(10, mu = q$center, Sigma = q$cov)
     Xnew$X1[,i] <- x[,1]
     Xnew$X2[,i] <- x[,2]
   }
qq <- mqcc(X, type = "T2", newdata = Xnew, pred.limits = TRUE)
summary(qq)

# generate new "out of control" data
Xnew <- list(X1 = matrix(NA, 10, 4), X2 =  matrix(NA, 10, 4))
for(i in 1:4)
   { x <- mvrnorm(10, mu = 1.2*q$center, Sigma = q$cov)
     Xnew$X1[,i] <- x[,1]
     Xnew$X2[,i] <- x[,2]
```

```
   }
qq <- mqcc(X, type = "T2", newdata = Xnew, pred.limits = TRUE)
summary(qq)

##
## Individual observations data
##

data(boiler)

q <- mqcc(boiler, type = "T2.single", confidence.level = 0.999)
summary(q)

# generate new "in control" data
boilerNew <- mvrnorm(10, mu = q$center, Sigma = q$cov)
qq <- mqcc(boiler, type = "T2.single", confidence.level = 0.999,
           newdata = boilerNew, pred.limits = TRUE)
summary(qq)

# generate new "out of control" data
boilerNew = mvrnorm(10, mu = 1.01*q$center, Sigma = q$cov)
qq <- mqcc(boiler, type = "T2.single", confidence.level = 0.999,
           newdata = boilerNew, pred.limits = TRUE)
summary(qq)

# provides "robust" estimates of means and covariance matrix
library(MASS)
rob <- cov.rob(boiler)
qrob <- mqcc(boiler, type = "T2.single", center = rob$center, cov = rob$cov)
summary(qrob)
```

`oc.curves` *Operating Characteristic Function*

Description

Draws the operating characteristic curves for a 'qcc' object.

Usage

```
oc.curves(object, ...)

oc.curves.xbar(object, n, c = seq(0, 5, length=101),
               nsigmas = object$nsigmas, identify=FALSE, restore.par=TRUE)

oc.curves.R(object, n, c = seq(1, 6, length=101),
            nsigmas = object$nsigmas, identify = FALSE, restore.par=TRUE)
```

```
oc.curves.S(object, n, c = seq(1, 6, length=101),
            nsigmas = object$nsigmas, identify = FALSE, restore.par=TRUE)

oc.curves.p(object, nsigmas = object$nsigmas, identify = FALSE, restore.par=TRUE)

oc.curves.c(object, nsigmas = object$nsigmas, identify = FALSE, restore.par=TRUE)
```

Arguments

object	an object of class 'qcc'.
identify	logical specifying whether to interactively identify points on the plot (see help for identify).
n	a vector of values specifying the sample sizes for which to draw the OC curves.
c	a vector of values specifying the multipliers for sigma in case of continuous variable.
nsigmas	a numeric value specifying the number of sigmas to use for computing control limits; if nsigmas is NULL, object$conf is used to set up probability limits; nsigmas is ignored for types "p" and "c".
restore.par	a logical value indicating whether the previous par settings must be restored. If you need to add points, lines, etc. to a chart set this to FALSE.
...	additional arguments to be passed to the generic function.

Details

An operating characteristic curve graphically provides information about the probability of not detecting a shift in the process. oc.curves is a generic function which calls the proper function depending on the type of 'qcc' object. Further arguments provided through ... are passed to the specific function depending on the type of chart.

The probabilities are based on the conventional assumptions about process distributions: the normal distribution for "xbar" , "R", and "S", the binomial distribution for "p" and "np", and the Poisson distribution for "c" and "u". They are all sensitive to departures from those assumptions, but to varying degrees. The performance of the "S" chart, and especially the "R" chart, are likely to be seriously affected by longer tails.

Value

The function invisibly returns a matrix or a vector of beta values, the probability of type II error.

Author(s)

Luca Scrucca

References

Mason, R.L. and Young, J.C. (2002) *Multivariate Statistical Process Control with Industrial Applications*, SIAM.

Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.

Ryan, T. P. (2000), *Statistical Methods for Quality Improvement*, 2nd ed. New York: John Wiley & Sons, Inc.
Scrucca, L. (2004). qcc: an R package for quality control charting and statistical process control. *R News* 4/1, 11-17.
Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

See Also

qcc

Examples

```
data(pistonrings)
attach(pistonrings)
diameter <- qcc.groups(diameter, sample)
beta <- oc.curves.xbar(qcc(diameter, type="xbar", nsigmas=3, plot=FALSE))
print(round(beta, digits=4))
# or to identify points on the plot use
## Not run: oc.curves.xbar(qcc(diameter, type="xbar", nsigmas=3, plot=FALSE), identify=TRUE)
detach(pistonrings)

data(orangejuice)
attach(orangejuice)
beta <- oc.curves(qcc(D[trial], sizes=size[trial], type="p", plot=FALSE))
print(round(beta, digits=4))
# or to identify points on the plot use
## Not run: oc.curves(qcc(D[trial], sizes=size[trial], type="p", plot=FALSE), identify=TRUE)
detach(orangejuice)

data(circuit)
attach(circuit)
q <- qcc(x[trial], sizes=size[trial], type="c", plot=FALSE)
beta <- oc.curves(q)
print(round(beta, digits=4))
# or to identify points on the plot use
## Not run: oc.curves(qcc(x[trial], sizes=size[trial], type="c", plot=FALSE), identify=TRUE)
detach(circuit)
```

---

orangejuice *Orange juice data*

Description

Frozen orange juice concentrate is packed in 6-oz cardboard cans. These cans are formed on a machine by spinning them from cardboard stock and attaching a metal bottom panel. A can is then inspected to determine whether, when filled, the liquid could possible leak either on the side seam or around the bottom joint. If this occurs, a can is considered nonconforming. The data were collected as 30 samples of 50 cans each at half-hour intervals over a three-shift period in which the machine was in continuous operation. From sample 15 used a new batch of cardboard stock was punt into

production. Sample 23 was obtained when an inexperienced operator was temporarily assigned to the machine. After the first 30 samples, a machine adjustment was made. Then further 24 samples were taken from the process.

Usage

```
data(orangejuice)
```

Format

A data frame with 54 observations on the following 4 variables:

sample sample id

D number of defectives

size sample sizes

trial trial samples (TRUE/FALSE)

References

Montgomery, D.C. (1991) *Introduction to Statistical Quality Control*, 2nd ed, New York, John Wiley & Sons, pp. 152–155.

Examples

```
data(orangejuice)
orangejuice$d <- orangejuice$D/orangejuice$size
attach(orangejuice)
summary(orangejuice)
boxplot(d ~ trial)
mark <- ifelse(trial, 1, 2)
plot(sample, d, type="b", col=mark, pch=mark)
detach(orangejuice)
```

---

orangejuice2 *Orange juice data – Part 2*

Description

A full description of the problem is given in orangejuice.

This dataset contains samples taken after the machine adjustment was made.

Usage

```
data(orangejuice)
```

Format

A data frame with 64 observations on the following 4 variables:

sample sample id

D number of defectives

size sample sizes

trial trial samples (TRUE/FALSE)

References

Montgomery, D.C. (1991) *Introduction to Statistical Quality Control*, 2nd ed, New York, John Wiley & Sons, pp. 155–159.

Examples

```
data(orangejuice2)
orangejuice2$d <- orangejuice2$D/orangejuice2$size
attach(orangejuice2)
summary(orangejuice2)
boxplot(d ~ trial)
mark <- ifelse(trial, 1, 2)
plot(sample, d, type="b", col=mark, pch=mark)
detach(orangejuice2)
```

`pareto.chart` *Pareto chart*

Description

Computes a table of statistics and plot a Pareto chart.

Usage

```
pareto.chart(data, plot = TRUE, ...)

## S3 method for class 'pareto.chart'
plot(x, xlab = NULL, ylab = "Frequency",
     ylab2 = "Cumulative Percentage",
     cumperc = seq(0, 100, by = 25),
     ylim = NULL, main = NULL,
     col = blues.colors(nlevels),
     ...)
```

Arguments

data	a vector of values. names(data) are used for labelling the bars.
plot	a logical specifying if the chart should be provided (TRUE, default).
x	the object of class 'pareto.chart' returned by a call to pareto.chart function.
xlab	a string specifying the label for the x-axis.
ylab	a string specifying the label for the y-axis.
ylab2	a string specifying the label for the second y-axis on the right side.
cumperc	a vector of percentage values to be used as tickmarks for the second y-axis on the right side.
ylim	a numeric vector specifying the limits for the y-axis.
main	a string specifying the main title to appear on the plot.
col	a value for the color, a vector of colors, or a palette for the bars. See the help for colors and palette.
...	other graphical arguments to be passed to the corresponding plot method, and eventually to the barplot function.

Details

A Pareto chart is a barplot where the categories are ordered in non increasing order, and a line is also added to show the cumulative sum.

Value

Returns an object of class 'pareto.chart' containing the descriptive statistics used to draw the Pareto chart. This object has associated a print and plot mehod.

Author(s)

Luca Scrucca

References

Mason, R.L. and Young, J.C. (2002) *Multivariate Statistical Process Control with Industrial Applications*, SIAM.
Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Ryan, T. P. (2000), *Statistical Methods for Quality Improvement*, 2nd ed. New York: John Wiley & Sons, Inc.
Scrucca, L. (2004). qcc: an R package for quality control charting and statistical process control. *R News* 4/1, 11-17.
Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

See Also

barplot

Examples

```
defect <- c(80, 27, 66, 94, 33)
names(defect) <- c("price code", "schedule date", "supplier code", "contact num.", "part num.")
pareto.chart(defect, ylab = "Error frequency")
pareto.chart(defect, ylab = "Error frequency", xlab = "Error causes", las=1)
pareto.chart(defect, ylab = "Error frequency", col=rainbow(length(defect)))
pareto.chart(defect, cumperc = seq(0, 100, by = 5), ylab2 = "A finer tickmarks grid")
```

---

pcmanufact *Personal computer manufacturer data*

Description

A personal computer manufacturer counts the number of nonconformities per unit on the final assembly line. He collects data on 20 samples of 5 computers each.

Usage

```
data(pcmanufact)
```

Format

A data frame with 10 observations on the following 2 variables.

x number of nonconformities (inspection units)

size number of computers inspected

References

Montgomery, D.C. (1991) *Introduction to Statistical Quality Control*, 2nd ed, New York, John Wiley & Sons, pp. 181–182

Examples

```
data(pcmanufact)
attach(pcmanufact)
summary(pcmanufact)
plot(x/size, type="b")
detach(pcmanufact)
```

pistonrings *Piston rings data*

Description

Piston rings for an automotive engine are produced by a forging process. The inside diameter of the rings manufactured by the process is measured on 25 samples, each of size 5, for the control phase I, when preliminary samples from a process being considered 'in control' are used to construct control charts. Then, further 15 samples, again each of size 5, are obtained for phase II.

Usage

```
data(pistonrings)
```

Format

A data frame with 200 observations on the following 3 variables.

diameter a numeric vector

sample sample ID

trial preliminary sample indicator (TRUE/FALSE)

References

Montgomery, D.C. (1991) *Introduction to Statistical Quality Control*, 2nd ed, New York, John Wiley & Sons, pp. 206–213

Examples

```
data(pistonrings)
attach(pistonrings)
summary(pistonrings)
boxplot(diameter ~ sample)
plot(sample, diameter, cex=0.7)
lines(tapply(diameter,sample,mean))
detach(pistonrings)
```

process.capability *Process capability analysis*

Description

Computes process capability indices for a 'qcc' object of type `"xbar"` and plot the histogram.

Usage

```
process.capability(object,spec.limits, target, std.dev, nsigmas,
                   confidence.level = 0.95, breaks = "scott",
                   add.stats = TRUE, print = TRUE,
                   digits = getOption("digits"), restore.par = TRUE)
```

Arguments

object	a 'qcc' object of type "xbar"
spec.limits	a two-values vector specifying the lower and upper specification limits. For one-sided specification limits, the value of the missing limit must be set to NA.
target	a value specifying the target of the process. If missing the value from the 'qcc' object is used if not NULL, otherwise the target is set at the middle value bewteen specification limits.
std.dev	a value specifying the within-group standard deviation. If not provided is taken from the 'qcc' object.
nsigmas	a numeric value specifying the number of sigmas to use. If not provided is taken from the 'qcc' object.
confidence.level	a numeric value between 0 and 1 specifying the level to use for computing confidence intervals.
breaks	a value or string used to draw the histogram. See the help for hist for more details.
add.stats	a logical value indicating whether statistics and capability indices should be added at the bottom of the chart.
print	a logical value indicating whether statistics and capability indices should be printed.
digits	the number of significant digits to use.
restore.par	a logical value indicating whether the previous par settings must be restored. If you need to add points, lines, etc. to a chart set this to FALSE.

Details

This function calculates confidence limits for C_p using the method described by Chou et al. (1990). Approximate confidence limits for C_{pl}, C_{pu} and C_{pk} are computed using the method in Bissell (1990). Confidence limits for C_{pm} are based on the method of Boyles (1991); this method is approximate and it assumes that the target is midway between the specification limits.

Value

Invisibly returns a list with components:

nobs	number of observations
center	center
std.dev	standard deviation

target	target
spec.limits	a vector of values giving the lower specification limit (LSL) and the upper specification limit (USL)
indices	a matrix of capability indices (C_p, C_{pl}, C_{pu}, C_{pk}, C_{pm}) and the corresponding confidence limits.
exp	a vector of values giving the expected fraction, based on a normal approximation, of the observations less than LSL and greater than USL.
obs	a vector of values giving the fraction of observations less than LSL and greater than USL.

Author(s)

Luca Scrucca

References

Bissell, A.F. (1990) *How reliable is your capability index?*, Applied Statistics, 39, 331-340.
Boyles, R.A. (1991) *The Taguchi capability index*, Journal of Quality Technology, 23, 107-126.
Chou, Y., Owen D.B. and Borrego S.A. (1990) *Lower Confidence Limits on Process Capability Indices*, Journal of Quality Technology, 22, 223-229.
Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

See Also

qcc

Examples

```
data(pistonrings)
attach(pistonrings)
diameter <- qcc.groups(diameter, sample)
q <- qcc(diameter[1:25,], type="xbar", nsigmas=3, plot=FALSE)
process.capability(q, spec.limits=c(73.95,74.05))
process.capability(q, spec.limits=c(73.95,74.05), target=74.02)
process.capability(q, spec.limits=c(73.99,74.01))
process.capability(q, spec.limits = c(73.99, 74.1))
```

---

qcc *Quality Control Charts*

Description

Create an object of class 'qcc' to perform statistical quality control. This object may then be used to plot Shewhart charts, drawing OC curves, computes capability indices, and more.

Usage

```
qcc(data, type, sizes, center, std.dev, limits,
    data.name, labels, newdata, newsizes, newdata.name,
    newlabels, nsigmas = 3, confidence.level,
    rules = shewhart.rules, plot = TRUE, ...)

## S3 method for class 'qcc'
print(x, ...)

## S3 method for class 'qcc'
summary(object, digits = getOption("digits"), ...)

## S3 method for class 'qcc'
plot(x, add.stats = TRUE, chart.all = TRUE,
     label.limits = c("LCL ", "UCL"), title, xlab, ylab, ylim,
     axes.las = 0, digits = getOption("digits"),
     restore.par = TRUE, ...)
```

Arguments

`data` a data frame, a matrix or a vector containing observed data for the variable to chart. Each row of a data frame or a matrix, and each value of a vector, refers to a sample or "rationale group".

`type` a character string specifying the group statistics to compute.
Available methods are:

	Statistic charted	Chart description
`"xbar"`	mean	means of a continuous process variable
`"R"`	range	ranges of a continuous process variable
`"S"`	standard deviation	standard deviations of a continuous variable
`"xbar.one"`	mean	one-at-time data of a continuous process variable
`"p"`	proportion	proportion of nonconforming units
`"np"`	count	number of nonconforming units
`"c"`	count	nonconformities per unit
`"u"`	count	average nonconformities per unit
`"g"`	count	number of non-events between events

Furthermore, a user specified type of chart, say `"newchart"`, can be provided. This requires the definition of `"stats.newchart"`, `"sd.newchart"`, and `"limits.newchart"`. As an example, see `stats.xbar`.

`sizes` a value or a vector of values specifying the sample sizes associated with each group. For continuous data provided as data frame or matrix the sample sizes are obtained counting the non-`NA` elements of each row. For `"p"`, `"np"` and `"u"` charts the argument `sizes` is required.

`center` a value specifying the center of group statistics or the "target" value of the process.

std.dev	a value or an available method specifying the within-group standard deviation(s) of the process. Several methods are available for estimating the standard deviation in case of a continuous process variable; see sd.xbar, sd.xbar.one, sd.R, sd.S.
limits	a two-values vector specifying control limits.
data.name	a string specifying the name of the variable which appears on the plots. If not provided is taken from the object given as data.
labels	a character vector of labels for each group.
newdata	a data frame, matrix or vector, as for the data argument, providing further data to plot but not included in the computations.
newsizes	a vector as for the sizes argument providing further data sizes to plot but not included in the computations.
newdata.name	a string specifying the name of the variable which appears on the plots. If not provided is taken from the object given as newdata.
newlabels	a character vector of labels for each new group defined in the argument newdata.
nsigmas	a numeric value specifying the number of sigmas to use for computing control limits. It is ignored when the confidence.level argument is provided.
confidence.level	a numeric value between 0 and 1 specifying the confidence level of the computed probability limits.
rules	a function of rules to apply to the chart. By default, the shewhart.rules function is used.
plot	logical. If TRUE a Shewhart chart is plotted.
add.stats	a logical value indicating whether statistics and other information should be printed at the bottom of the chart.
chart.all	a logical value indicating whether both statistics for data and for newdata (if given) should be plotted.
label.limits	a character vector specifying the labels for control limits.
title	a string giving the label for the main title.
xlab	a string giving the label for the x-axis.
ylab	a string giving the label for the y-axis.
ylim	a numeric vector specifying the limits for the y-axis.
axes.las	numeric in {0,1,2,3} specifying the style of axis labels. See help(par).
digits	the number of significant digits to use.
restore.par	a logical value indicating whether the previous par settings must be restored. If you need to add points, lines, etc. to a control chart set this to FALSE.
object	an object of class 'qcc'.
x	an object of class 'qcc'.
...	additional arguments to be passed to the generic function.

Value

Returns an object of class 'qcc'.

Note

For a nice blog post discussing the qcc package, in particular how to implement the *Western Eletric Rules (WER)*, see http://blog.yhathq.com/posts/quality-control-in-r.html.

Author(s)

Luca Scrucca

References

Mason, R.L. and Young, J.C. (2002) *Multivariate Statistical Process Control with Industrial Applications*, SIAM.
Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Ryan, T. P. (2000), *Statistical Methods for Quality Improvement*, 2nd ed. New York: John Wiley & Sons, Inc.
Scrucca, L. (2004). qcc: an R package for quality control charting and statistical process control. *R News* 4/1, 11-17.
Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

See Also

shewhart.rules, cusum, ewma, oc.curves, process.capability, qcc.groups

Examples

```
##
##  Continuous data
##
data(pistonrings)
attach(pistonrings)
diameter <- qcc.groups(diameter, sample)

qcc(diameter[1:25,], type="xbar")
qcc(diameter[1:25,], type="xbar", newdata=diameter[26:40,])
q <- qcc(diameter[1:25,], type="xbar", newdata=diameter[26:40,], plot=FALSE)
plot(q, chart.all=FALSE)
qcc(diameter[1:25,], type="xbar", newdata=diameter[26:40,], nsigmas=2)
qcc(diameter[1:25,], type="xbar", newdata=diameter[26:40,], confidence.level=0.99)

qcc(diameter[1:25,], type="R")
qcc(diameter[1:25,], type="R", newdata=diameter[26:40,])

qcc(diameter[1:25,], type="S")
qcc(diameter[1:25,], type="S", newdata=diameter[26:40,])

# add warning limits at 2 std. deviations
q <- qcc(diameter[1:25,], type="xbar", newdata=diameter[26:40,], plot=FALSE)
(warn.limits <- limits.xbar(q$center, q$std.dev, q$sizes, 2))
plot(q, restore.par = FALSE)
abline(h = warn.limits, lty = 3, col = "chocolate")
```

```
# variable control limits
out <- c(9, 10, 30, 35, 45, 64, 65, 74, 75, 85, 99, 100)
diameter <- qcc.groups(pistonrings$diameter[-out], sample[-out])
qcc(diameter[1:25,], type="xbar")
qcc(diameter[1:25,], type="R")
qcc(diameter[1:25,], type="S")
qcc(diameter[1:25,], type="xbar", newdata=diameter[26:40,])
qcc(diameter[1:25,], type="R", newdata=diameter[26:40,])
qcc(diameter[1:25,], type="S", newdata=diameter[26:40,])

detach(pistonrings)

##
##  Attribute data
##

data(orangejuice)
attach(orangejuice)
qcc(D[trial], sizes=size[trial], type="p")

# remove out-of-control points (see help(orangejuice) for the reasons)
inc <- setdiff(which(trial), c(15,23))
q1 <- qcc(D[inc], sizes=size[inc], type="p")
qcc(D[inc], sizes=size[inc], type="p", newdata=D[!trial], newsizes=size[!trial])
detach(orangejuice)

data(orangejuice2)
attach(orangejuice2)
names(D) <- sample
qcc(D[trial], sizes=size[trial], type="p")
q2 <- qcc(D[trial], sizes=size[trial], type="p", newdata=D[!trial], newsizes=size[!trial])
detach(orangejuice2)

# put on the same graph the two orange juice samples
oldpar <- par(no.readonly = TRUE)
par(mfrow=c(1,2), mar=c(5,5,3,0))
plot(q1, title="First samples", ylim=c(0,0.5), add.stats=FALSE, restore.par=FALSE)
par("mar"=c(5,0,3,3), yaxt="n")
plot(q2, title="Second samples", add.stats=FALSE, ylim=c(0,0.5))
par(oldpar)

data(circuit)
attach(circuit)
qcc(x[trial], sizes=size[trial], type="c")
# remove out-of-control points (see help(circuit) for the reasons)
inc <- setdiff(which(trial), c(6,20))
qcc(x[inc], sizes=size[inc], type="c", labels=inc)
qcc(x[inc], sizes=size[inc], type="c", labels=inc,
    newdata=x[!trial], newsizes=size[!trial], newlabels=which(!trial))
qcc(x[inc], sizes=size[inc], type="u", labels=inc,
    newdata=x[!trial], newsizes=size[!trial], newlabels=which(!trial))
detach(circuit)
```

```
data(pcmanufact)
attach(pcmanufact)
qcc(x, sizes=size, type="u")
detach(pcmanufact)

data(dyedcloth)
attach(dyedcloth)
qcc(x, sizes=size, type="u")
# standardized control chart
q <- qcc(x, sizes=size, type="u", plot=FALSE)
z <- (q$statistics - q$center)/sqrt(q$center/q$size)
plot(z,  type="o", ylim=range(z,3,-3), pch=16)
abline(h=0, lty=2)
abline(h=c(-3,3), lty=2)
detach(dyedcloth)

##
##  Continuous one-at-time data
##

# viscosity data (Montgomery, pag. 242)
x <- c(33.75, 33.05, 34, 33.81, 33.46, 34.02, 33.68, 33.27, 33.49, 33.20,
       33.62, 33.00, 33.54, 33.12, 33.84)
qcc(x, type="xbar.one")
qcc(x, type="xbar.one", std.dev = "SD")
```

`qcc.groups` *Grouping data based on a sample indicator*

Description

This function allows to easily group data to use as input to the 'qcc' function.

Usage

```
qcc.groups(data, sample)
```

Arguments

`data`	the observed data values
`sample`	the sample indicators for the data values

Value

The function returns a matrix of suitable dimensions. If one or more group have few observations than others, `NA` values are appended.

Author(s)

Luca Scrucca

See Also

qcc

Examples

```
data(pistonrings)
attach(pistonrings)
# 40 sample of 5 obs each
qcc.groups(diameter, sample)
# some obs are removed, the result is still a 40x5 matrix but with NAs added
qcc.groups(diameter[-c(1,2,50,52, 199)], sample[-c(1,2,50,52, 199)])
```

---

qcc.options *Set or return options for the 'qcc' package.*

Description

This function can be used to control the behavior of the 'qcc' library such as the background color, out-of-control points appearance, and many others.

Usage

```
qcc.options(...)
```

Arguments

...	the option to be set or retrieved. See details.

Details

The available options are:

`exp.R.unscaled` a vector specifying, for each sample size, the expected value of the relative range (i.e. R/σ) for a normal distribution. This appears as d_2 on most tables containing factors for the construction of control charts.

`se.R.unscaled` a vector specifying, for each sample size, the standard error of the relative range (i.e. R/σ) for a normal distribution. This appears as d_3 on most tables containing factors for the construction of control charts.

`beyond.limits$pch` plotting character used to highlight points beyond control limits.

`beyond.limits$col` color used to highlight points beyond control limits.

`violating.runs$pch` plotting character used to highlight points violating runs.

`violating.runs$col` color used to highlight points violating runs.

`run.length` the maximum value of a run before to signal a point as out of control.

`bg.margin` background color used to draw the margin of the charts.

`bg.figure` background color used to draw the figure of the charts.

`cex` character expansion used to draw plot annotations (labels, title, tickmarks, etc.).

`font.stats` font used to draw text at the bottom of control charts.

`cex.stats` character expansion used to draw text at the bottom of control charts.

Value

If the functions is called with no argument return a list of available options.

If an option argument is provided the corresponding value is returned.

If a value is associated with an option argument, such option is set and the list of updated option values is invisibly returned. In this case the list `.qcc.options` is modified and any modification will remain in effect for the rest of the session.

Author(s)

Luca Scrucca

See Also

qcc

Examples

```
old <- qcc.options() # save defaults
qcc.options("cex.stats") # get a single parameter
qcc.options("cex.stats"=1.2) # change parameters
qcc.options(bg.margin="azure2")
qcc.options("violating.runs" = list(pch = 15, col = "purple"))
qcc.options("beyond.limits" = list(pch = 15, col = "orangered"))
qcc(rnorm(100), type = "xbar.one", std.dev = 0.7) # see the results
qcc.options(old) # restore old defaults
```

---

qcc.overdispersion.test

Overdispersion test for binomial and poisson data

Description

This function allows to test for overdispersed data in the binomial and poisson case.

Usage

```
qcc.overdispersion.test(x, size, type=ifelse(missing(size), "poisson", "binomial"))
```

Arguments

`x`	a vector of observed data values
`size`	for binomial data, a vector of sample sizes
`type`	a character string specifying the distribution for testing, either `"poisson"` or `"binomial"`. By default, if `size` is provided a binomial distributed is assumed, otherwise a poisson distribution.

Details

This very simple test amounts to compute the statistic

$$\text{D} = \text{Observed variance / Theoretical variance} \times (\text{no. observations - 1})$$

and refer this to a Chi-square distribution with (no. observations - 1) degrees of freedom.

Value

The function returns a matrix of results.

Author(s)

Luca Scrucca

References

Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*, New York, Chapman and Hall, pp. 216–218

Examples

```
# data from Wetherill and Brown (1991) pp. 212--213, 216--218:
x <- c(12,11,18,11,10,16,9,11,14,15,11,9,10,13,12,
       8,12,13,10,12,13,16,12,18,16,10,16,10,12,14)
size <- rep(50, length(x))
qcc.overdispersion.test(x,size)

x <- c(11,8,13,11,13,17,25,23,11,16,9,15,10,16,12,
       8,9,15,4,12,12,12,15,17,14,17,12,12,7,16)
qcc.overdispersion.test(x)
```

shewhart.rules *Functions specifying rules for Shewhart charts*

Description

These functions are used to signal out of control points in Shewhart charts.

Usage

```
shewhart.rules(object, limits = object$limits, run.length = qcc.options("run.length"))
beyond.limits(object, limits = object$limits)
violating.runs(object, run.length = qcc.options("run.length"))
```

Arguments

object	an object of class 'qcc'.
limits	control limits
run.length	the maximum value of a run before to signal a point as out of control.

Details

The function shewhart.rules simply calls the beyond.limits and violating.runs functions which actually do the real calculations.

Value

The shewhart.rules function returns a list with components:

beyond.limits	the indices of points beyond control limits.
violating.runs	the indices of points violating runs.

Author(s)

Luca Scrucca

stats.c *Functions to plot Shewhart c chart*

Description

Statistics used in computing and drawing a Shewhart c chart.

Usage

```
stats.c(data, sizes)
sd.c(data, sizes, ...)
limits.c(center, std.dev, sizes, conf)
```

Arguments

`data`	the observed data values
`center`	sample/group center statistic.
`sizes`	samples sizes.
`std.dev`	within group standard deviation.
`conf`	a numeric value used to compute control limits, specifying the number of standard deviations (if `conf` > 1) or the confidence level (if 0 < `conf` < 1).
`...`	further arguments are ignored.

Value

The function `stats.c` returns a list with components `statistics` and `center`.

The function `sd.c` returns `std.dev` the standard deviation of the statistic charted.

The function `limits.c` returns a matrix with lower and upper control limits.

Author(s)

Luca Scrucca

References

Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

See Also

`qcc`

`stats.g` *Statistics used in computing and drawing a Shewhart g chart*

Description

These functions are used to compute statistics required by the g chart (geometric distribution) for use with the qcc package.

Usage

```
stats.g(data, sizes)
sd.g(data, sizes, ...)
limits.g(center, std.dev, sizes, conf)
```

Arguments

data	the observed data values
center	sample center statistic
sizes	sample sizes (not used)
std.dev	standard deviation of geometric distribution
conf	a numeric value used to compute control limits, specifying the number of standard deviations (if 'conf' > 1) or the confidence level (if 0 < 'conf' < 1).
...	further arguments are ignored.

Details

The g chart plots the number of non-events between events. np charts do not work well when the probability of an event is rare (see example below). Instead of plotting the number of events, the g chart plots the number of non-events between events.

Value

The function `stats.g()` returns a list with components `statistics` and `center`.

The function `sd.g()` returns `std.dev` the standard deviation $sqrt(1-p)/p$.

The function `limits.g()` returns a matrix with lower and upper control limits.

Note

The geometric distribution is quite skewed so it is best to set conf at the required confidence interval (0 < conf < 1) rather than as a multiplier of sigma.

Author(s)

Greg Snow <greg.snow@ihc.com>

References

Kaminsky, FC et. al. (1992) *Statistical Control Charts Based on a Geometric Distribution*, Journal of Quality Technology, 24, pp 63–69.
Yang, Z et. al. (2002) On the Performance of Geometric Charts with Estimated Control Limits, *Journal of Quality Technology*, 34, pp 448–458.

See Also

`qcc`

Examples

```
success <- rbinom(1000, 1, 0.01)
num.noevent <- diff(which(c(1,success)==1))-1
qcc(success, type = "np", sizes = 1)
qcc(num.noevent, type = "g")
```

stats.np *Statistics used in computing and drawing a Shewhart np chart*

Description

These functions are used to compute statistics required by the np chart.

Usage

```
stats.np(data, sizes)
sd.np(data, sizes, ...)
limits.np(center, std.dev, sizes, conf)
```

Arguments

data	the observed data values
center	sample/group center statistic.
sizes	samples sizes.
std.dev	within group standard deviation.
conf	a numeric value used to compute control limits, specifying the number of standard deviations (if conf > 1) or the confidence level (if 0 < conf < 1).
...	further arguments are ignored.

Value

The function stats.np returns a list with components statistics and center.

The function sd.np returns std.dev the standard deviation of the statistic charted.

The function limits.np returns a matrix with lower and upper control limits.

Author(s)

Luca Scrucca

References

Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

See Also

qcc

`stats.p` *Statistics used in computing and drawing a Shewhart p chart*

Description

These functions are used to compute statistics required by the p chart.

Usage

```
stats.p(data, sizes)
sd.p(data, sizes, ...)
limits.p(center, std.dev, sizes, conf)
```

Arguments

`data`	the observed data values
`center`	sample/group center statistic.
`sizes`	samples sizes.
`std.dev`	within group standard deviation.
`conf`	a numeric value used to compute control limits, specifying the number of standard deviations (if `conf` > 1) or the confidence level (if 0 < `conf` < 1).
`...`	further arguments are ignored.

Value

The function `stats.p` returns a list with components `statistics` and `center`.

The function `sd.p` returns `std.dev` the standard deviation of the statistic charted.

The function `limits.p` returns a matrix with lower and upper control limits.

Author(s)

Luca Scrucca

References

Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

See Also

`qcc`

stats.R *Statistics used in computing and drawing a Shewhart R chart*

Description

These functions are used to compute statistics required by the R chart.

Usage

```
stats.R(data, sizes)
sd.R(data, sizes, std.dev = c("UWAVE-R", "MVLUE-R"))
limits.R(center, std.dev, sizes, conf)
```

Arguments

data	the observed data values
center	sample/group center statistic.
sizes	samples sizes. Optional
std.dev	within group standard deviation. Optional for sd.R function, required for limits.R. See sd.xbar.
conf	a numeric value used to compute control limits, specifying the number of standard deviations (if conf > 1) or the confidence level (if 0 < conf < 1).

Value

The function stats.R returns a list with components statistics and center.

The function sd.R returns std.dev the standard deviation of the statistic charted.

The function limits.R returns a matrix with lower and upper control limits.

Author(s)

Luca Scrucca

References

Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

See Also

qcc

stats.S *Functions to plot Shewhart S chart*

Description

These functions are used to compute statistics required by the S chart.

Usage

```
stats.S(data, sizes)
sd.S(data, sizes, std.dev = c("UWAVE-SD", "MVLUE-SD", "RMSDF"))
limits.S(center, std.dev, sizes, conf)
```

Arguments

data	the observed data values
center	sample/group center statistic.
sizes	samples sizes. Optional
std.dev	within group standard deviation. Optional for sd.S function, required for limits.S. See sd.xbar.
conf	a numeric value used to compute control limits, specifying the number of standard deviations (if conf > 1) or the confidence level (if 0 < conf < 1).

Value

The function stats.S returns a list with components statistics and center.

The function sd.S returns std.dev the standard deviation of the statistic charted.

The function limits.S returns a matrix with lower and upper control limits.

Author(s)

Luca Scrucca

References

Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

See Also

qcc

`stats.T2` *Statistics used in computing and drawing the Hotelling T^2 chart for subgrouped data*

Description

These functions are used to compute statistics required by the T^2 chart.

Usage

```
stats.T2(data, center = NULL, cov = NULL)

limits.T2(ngroups, size, nvars, conf)
```

Arguments

`data`	the observed data values
`center`	a vector of values to use for center of input variables.
`cov`	a matrix of values to use for the covariance matrix of input variables.
`ngroups`	number of groups
`size`	sample size
`nvars`	number of variables
`conf`	confidence level ($0 <$ `conf` < 1)

Value

The function `stats.T2` returns a list with components:

`statistics`	a vector of values for the T^2 statistic
`means`	a matrix of within group means for each variable
`center`	sample/group center statistic
`S`	covariance matrix

The function `limits.T2` returns a list with components:

`control`	control limits
`prediction`	pred.limits

Author(s)

Luca Scrucca

References

Mason, R.L. and Young, J.C. (2002) *Multivariate Statistical Process Control with Industrial Applications*, SIAM.
Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Ryan, T. P. (2000), *Statistical Methods for Quality Improvement*, 2nd ed. New York: John Wiley & Sons, Inc.

See Also

mqcc, stats.T2.single

`stats.T2.single` *Statistics used in computing and drawing the Hotelling T^2 chart for individual observations data*

Description

These functions are used to compute statistics required by the T^2 chart for individual observations.

Usage

```
stats.T2.single(data, center = NULL, cov = NULL)

limits.T2.single(ngroups, size, nvars, conf)
```

Arguments

`data`	the observed data values
`center`	a vector of values to use for center of input variables.
`cov`	a matrix of values to use for the covariance matrix of input variables.
`ngroups`	number of groups
`size`	sample size
`nvars`	number of variables
`conf`	confidence level (0 < `conf` < 1)

Value

The function `stats.T2.single` returns a list with components:

`statistics`	a vector of values for the T^2 statistic
`means`	a matrix of within group means for each variable (which is equal to `data` since sample are of sizes one)
`center`	sample/group center statistic
`S`	covariance matrix

The function `limits.T2.single` returns a list with components:

`control`	control limits
`prediction`	pred.limits

Author(s)

Luca Scrucca

References

Mason, R.L. and Young, J.C. (2002) *Multivariate Statistical Process Control with Industrial Applications*, SIAM.
Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Ryan, T. P. (2000), *Statistical Methods for Quality Improvement*, 2nd ed. New York: John Wiley & Sons, Inc.

See Also

`mqcc`, `stats.T2`

---

stats.u *Statistics used in computing and drawing a Shewhart u chart*

Description

These functions are used to compute statistics required by the u chart.

Usage

```
stats.u(data, sizes)
sd.u(data, sizes, ...)
limits.u(center, std.dev, sizes, conf)
```

Arguments

`data`	the observed data values
`center`	sample/group center statistic.
`sizes`	samples sizes.
`std.dev`	within group standard deviation.
`conf`	a numeric value used to compute control limits, specifying the number of standard deviations (if `conf` > 1) or the confidence level (if 0 < `conf` < 1).
`...`	further arguments are ignored.

Value

The function `stats.u` returns a list with components `statistics` and `center`.

The function `sd.u` returns `std.dev` the standard deviation of the statistic charted.

The function `limits.u` returns a matrix with lower and upper control limits.

Author(s)

Luca Scrucca

References

Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

See Also

qcc

`stats.xbar` *Statistics used in computing and drawing a Shewhart xbar chart*

Description

These functions are used to compute statistics required by the xbar chart.

Usage

```
stats.xbar(data, sizes)
sd.xbar(data, sizes, std.dev = c("UWAVE-R", "UWAVE-SD", "MVLUE-R", "MVLUE-SD", "RMSDF"))
limits.xbar(center, std.dev, sizes, conf)
```

Arguments

`data`	the observed data values
`center`	sample/group center statistic
`sizes`	samples sizes. Optional
`std.dev`	within group standard deviation. Optional for `sd.xbar` function, required for `limits.xbar`. See details.
`conf`	a numeric value used to compute control limits, specifying the number of standard deviations (if `conf` > 1) or the confidence level (if 0 < `conf` < 1).

Details

The following methods are available for estimating the process standard deviation:

`"UWAVE-R"` UnWeighted AVErage of subgroup estimates based on subgroup Ranges

`"UWAVE-SD"` UnWeighted AVErage of subgroup estimates based on subgroup Standard Deviations

`"MVLUE-R"` Minimum Variance Linear Unbiased Estimator computed as a weighted average of subgroups estimates based on subgroup Ranges

`"MVLUE-SD"` Minimum Variance Linear Unbiased Estimator computed as a weighted average of subgroup estimates based on subgroup Standard Deviations

`"RMSDF"` Root-Mean-Square estimator computed as a weighted average of subgroup estimates based on subgroup Standard Deviations

Depending on the chart, a method may be available or not, or set as the default according to the following table:

Method	`"xbar"`	`"R"`	`"S"`
`"UWAVE-R"`	default	default	not available
`"UWAVE-SD"`		not available	default
`"MVLUE-R"`			not available
`"MVLUE-SD"`		not available	
`"RMSDF"`		not available	

Detailed definitions of formulae implemented are available in the SAS/QC 9.2 User's Guide.

Value

The function `stats.xbar` returns a list with components `statistics` and `center`.

The function `sd.xbar` returns `std.dev` the standard deviation of the statistic charted. This is based on results from Burr (1969).

The function `limits.xbar` returns a matrix with lower and upper control limits.

Author(s)

Luca Scrucca

References

Burr, I.W. (1969) Control charts for measurements with varying sample sizes. *Journal of Quality Technology*, 1(3), 163-167.

Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.

Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

See Also

`qcc`

`stats.xbar.one` *Statistics used in computing and drawing a Shewhart xbar chart for one-at-time data*

Description

These functions are used to compute statistics required by the xbar chart for one-at-time data.

Usage

```
stats.xbar.one(data, sizes)
sd.xbar.one(data, sizes, std.dev = c("MR", "SD"), k=2)
limits.xbar.one(center, std.dev, sizes, conf)
```

Arguments

`data`	the observed data values
`center`	sample/group center statistic.
`sizes`	samples sizes. Not needed, size=1 is used.
`k`	number of successive pairs of observations for computing the standard deviation based on moving ranges of k points.
`std.dev`	within group standard deviation. Optional for `sd.xbar.one` function, required for `limits.xbar.one`. See details.
`conf`	a numeric value used to compute control limits, specifying the number of standard deviations (if `conf` > 1) or the confidence level (if 0 < `conf` < 1).

Details

Methods available for estimating the process standard deviation:

Method	Description
`"MR"`	moving range: this is estimate is based on the scaled mean of moving ranges
`"SD"`	sample standard deviation: this estimate is defined as as(x)/cd(n), where n = number of observations x.

Value

The function `stats.xbar.one` returns a list with components `statistics` and `center`.

The function `sd.xbar.one` returns `std.dev` the standard deviation of the statistic charted.

The function `limits.xbar.one` returns a matrix with lower and upper control limits.

Author(s)

Luca Scrucca

References

Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed. New York: John Wiley & Sons.
Ryan T.P. (2000) *Statistical Methods for Quality Improvement*, New York: John Wiley & Sons.
Wetherill, G.B. and Brown, D.W. (1991) *Statistical Process Control*. New York: Chapman & Hall.

See Also

qcc

Examples

```
# Water content of antifreeze data (Wetherill and Brown, 1991, p. 120)
x <- c(2.23, 2.53, 2.62, 2.63, 2.58, 2.44, 2.49, 2.34, 2.95, 2.54, 2.60, 2.45,
       2.17, 2.58, 2.57, 2.44, 2.38, 2.23, 2.23, 2.54, 2.66, 2.84, 2.81, 2.39,
       2.56, 2.70, 3.00, 2.81, 2.77, 2.89, 2.54, 2.98, 2.35, 2.53)
# the Shewhart control chart for one-at-time data
# 1) using MR (default)
qcc(x, type="xbar.one", data.name="Water content (in ppm) of batches of antifreeze")
# 2) using SD
qcc(x, type="xbar.one", std.dev = "SD", data.name="Water content (in ppm) of batches of antifreeze")

# "as the size increases further, we would expect sigma-hat to settle down
#  at a value close to the overall sigma-hat" (Wetherill and Brown, 1991,
# p. 121)
sigma <- NA
k <- 2:24
for (j in k)
    sigma[j] <- sd.xbar.one(x, k=j)
plot(k, sigma[k], type="b")     # plot estimates of sigma for
abline(h=sd(x), col=2, lty=2)   # different values of k
```

Index