

Package: qcc (via r-universe)

October 8, 2026

Version 3.0

Date 2026-09-08

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 (>= 4.4), ggplot2 (>= 4.0), patchwork (>= 1.3)

Imports graphics, grDevices, grid, MASS, scales, cli, crayon

Suggests knitr (>= 1.5), rmarkdown (>= 2.3), testthat (>= 3.0.0), vdiffr (>= 1.0.0)

VignetteBuilder knitr

License GPL (>= 2)

URL <https://luca-scr.github.io/qcc/>

BugReports <https://github.com/luca-scr/qcc/issues>

ByteCompile true

LazyData true

Encoding UTF-8

Config/testthat/edition 3

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.0.0

Repository <https://luca-scr.r-universe.dev>

Date/Publication 2026-09-08 09:21:20 UTC

RemoteUrl <https://github.com/luca-scr/qcc>

RemoteRef HEAD

RemoteSha e09918a0a02f4ff95859135943f8211a048e35a0

Contents

antifreeze	3
blues.colors	3
boiler	4
c4	5
causeEffectDiagram	5
circuit	7
cusum	7
d2	11
d3	12
describe	13
dyedcloth	14
ellipseChart	14
ewma	16
ewmaSmooth	19
mqcc	20
nclass.hist	24
ocCurves	25
orangejuice	27
paretoChart	28
pmanufact	30
pistonrings	31
processCapability	32
qcc	34
qcc-deprecated	40
qcc.options	40
qccGroups	41
qccOverdispersionTest	42
rules	43
RyanMultivar	45
stats.c	46
stats.g	47
stats.np	48
stats.p	49
stats.R	50
stats.S	51
stats.T2	52
stats.T2.single	53
stats.u	55
stats.xbar	56
stats.xbar.one	58
viscosity	60

Index

antifreeze	<i>Water content of antifreeze data</i>
------------	---

Description

Water content in 34 successive batches of antifreeze.

Usage

```
antifreeze
```

Format

A data frame with 34 rows and 1 column:

water water content, in ppm

Source

Wetherill, G.B. and Brown, D.W. (1991) Statistical Process Control. New York: Chapman & Hall, p. 120

See Also

Other datasets in qcc package: [RyanMultivar](#), [boiler](#), [circuit](#), [dyedcloth](#), [orangejuice](#), [pcmanufact](#), [pistonrings](#), [viscosity](#)

blues.colors	<i>Blue color palette</i>
--------------	---------------------------

Description

Creates a palette of blue colors.

Usage

```
blues.colors(n)
```

Arguments

n the number of colors.

Value

A character vector containing the requested colors.

boiler	<i>Boiler temperature data</i>
--------	--------------------------------

Description

Temperature readings from the eight configured burners on a boiler.

Usage

```
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

Source

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

See Also

Other datasets in qcc package: [RyanMultivar](#), [antifreeze](#), [circuit](#), [dyedcloth](#), [orangejuice](#), [pcmanufact](#), [pistonrings](#), [viscosity](#)

c4	<i>The c_4 Constant</i>
----	-------------------------

Description

Calculates the bias-correction factor for the sample standard deviation.

Usage

```
c4(n)
```

Arguments

n A vector of sample sizes.

Details

For a normally distributed sample of size n , the sample standard deviation S is biased downward, with

$$E(S) = c_4(n)\sigma.$$

Consequently, $S/c_4(n)$ is an unbiased estimator of the population standard deviation.

Value

A vector of calculated c_4 constants.

See Also

For other implementations in R: [SixSigma::ss.cc.getc4\(\)](#), [IQCC::c4\(\)](#), [shewhartr::shewhart_constants\(\)](#)
 Other constants for Shewhart charts: [d2\(\)](#), [d3\(\)](#)

causeEffectDiagram	<i>Cause and Effect Diagram</i>
--------------------	---------------------------------

Description

Draw a basic cause and effect diagram.

Usage

```
causeEffectDiagram(
  cause,
  effect,
  title = "Cause-and-Effect diagram",
  cex = c(0.9, 1, 1.2),
  font = c(3, 1, 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 character string specifying the main title. Set title = NULL to remove the title.
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.
...	catches further ignored arguments.

Author(s)

Luca Scrucca

References

- Montgomery, D.C. (2013). *Introduction to Statistical Quality Control*, 7th ed. New York: John Wiley & Sons.
- Wetherill, G. B., and Brown, D. W. (1991). *Statistical Process Control*. New York: Chapman & Hall.

Examples

```
causeEffectDiagram(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

```
circuit
```

Format

A data frame with 46 observations on the following 4 variables:

sample sample number

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

size sample size

trial trial sample indicator (TRUE/FALSE)

Source

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

See Also

Other datasets in qcc package: [RyanMultivar](#), [antifreeze](#), [boiler](#), [dyedcloth](#), [orangejuice](#), [pcmanufact](#), [pistonrings](#), [viscosity](#)

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,
  decision.interval = 5,
  se.shift = 1,
  head.start = 0,
  newdata,
  newsizes,
  ...
)

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

## S3 method for class 'cusum.qcc'
summary(object, ...)

## S3 method for class 'cusum.qcc'
plot(
  x,
  xtime = NULL,
  add.stats = getOption("qcc.add.stats"),
  chart.all = getOption("qcc.chart.all"),
  fill = getOption("qcc.fill"),
  label.bounds = c("LDB", "UDB"),
  title,
  xlab,
  ylab,
  xlim,
  ylim,
  digits = getOption("digits"),
  ...
)

```

Arguments

<code>data</code>	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".
<code>sizes</code>	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.
<code>center</code>	a value specifying the center of group statistics or target.

<code>std.dev</code>	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 <code>sd.xbar()</code> and <code>sd.xbar.one()</code> for, respectively, the grouped data case and the individual observations case.
<code>decision.interval</code>	A numeric value specifying the number of standard errors of the summary statistics at which the cumulative sum is out of control.
<code>se.shift</code>	The amount of shift to detect in the process, measured in standard errors of the summary statistics.
<code>head.start</code>	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 <code>decision.interval</code> for a Fast Initial Response.
<code>newdata</code>	a data frame, matrix or vector, as for the <code>data</code> argument, providing further data to plot but not included in the computations.
<code>newsizes</code>	a vector as for the <code>sizes</code> argument providing further data sizes to plot but not included in the computations.
<code>...</code>	additional arguments to be passed to the generic function.
<code>x</code>	an object of class <code>'cusum.qcc'</code> .
<code>digits</code>	the number of significant digits to use.
<code>object</code>	an object of class <code>'cusum.qcc'</code> .
<code>xtime</code>	a vector of date-time values as returned by <code>Sys.time()</code> and <code>Sys.Date()</code> . If provided it is used for x-axis so it must be of the same length as the statistic charted.
<code>add.stats</code>	a logical value indicating whether statistics and other information should be printed at the bottom of the chart.
<code>chart.all</code>	a logical value indicating whether both statistics for data and for <code>newdata</code> (if given) should be plotted.
<code>fill</code>	a logical value specifying if the in-control area should be filled with the color specified in the <code>qcc.zones</code> option.
<code>label.bounds</code>	a character vector specifying the labels for the the decision interval boundaries.
<code>title</code>	a character string specifying the main title. Set <code>title = NULL</code> to remove the title.
<code>xlab</code>	a string giving the label for the x-axis.
<code>ylab</code>	a string giving the label for the y-axis.
<code>xlim</code>	a numeric vector specifying the limits for the x-axis.
<code>ylim</code>	a numeric vector specifying the limits for the y-axis.

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. (2013). *Introduction to Statistical Quality Control*, 7th ed. New York: John Wiley & Sons.
- Ryan, T. P. (2011). *Statistical Methods for Quality Improvement*, 3rd 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

Other control charts: [ewma\(\)](#), [qcc\(\)](#)

Examples

```
## Grouped-data
diameter <- qccGroups(data = pistonrings, diameter, sample)

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

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

## Individual observations
q <- with(viscosity, cusum(viscosity[trial], newdata = viscosity[!trial]))
summary(q)
plot(q)
```

d2	<i>The d_2 Constant</i>
----	-------------------------

Description

Calculate the d_2 constants for a vector n of sample sizes.

Usage

`d2(n)`

Arguments

n A vector of sample sizes.

Details

$d_2(n)$ is the expected value of the range of n observations from a standard normal distribution.

$$d_2(n) = \frac{E(r)}{\sigma}$$

Value

A vector of calculated d_2 constants.

References

- Cano, Emilio L., Moguerza, Javier M. and Redchuk, Andres. 2012. *Six Sigma with R. Statistical Engineering for ProcessImprovement*, Use R!, vol. 36. Springer, New York.

See Also

For other implementations in R: `SixSigma::ss.cc.getd2()`, `IQCC::d2()`, `shewhartr::shewhart_constants()`

Other constants for Shewhart charts: `c4()`, `d3()`

d3

*The d_3 Constant***Description**

Calculate the d_3 constants for a vector n of sample sizes.

Usage

`d3(n)`

Arguments

`n` A vector of sample sizes.

Details

$d_3(n)$ is the standard deviation of the range of n observations from a standard normal distribution.

$$d_3(n) = \frac{Stddev(r)}{\sigma}$$

Not to be confused with the D_3 , which is

$$D_4 = 1 - 3 \frac{d_3}{d_2}$$

Value

A vector of calculated d_3 constants.

References

- Cano, Emilio L., Moguerza, Javier M. and Redchuk, Andres. 2012. *Six Sigma with R. Statistical Engineering for ProcessImprovement*, Use R!, vol. 36. Springer, New York.

See Also

For other implementations in R: `SixSigma::ss.cc.getd3()`, `IQCC::d3()`, `shewhartr::shewhart_constants()`

Other constants for Shewhart charts: `c4()`, `d2()`

describe	<i>Descriptive statistics</i>
----------	-------------------------------

Description

Compute and print descriptive statistics for a matrix or data frame, eventually conditioning on another variable.

Usage

```
describe(data, by, detailed = FALSE, ...)  
  
## S3 method for class 'describe'  
print(x, digits = getOption("digits") - 3, ...)
```

Arguments

data	a matrix or data frame.
by	a factor or character vector specifying the conditioning variable. This may be a variable included in data or an object defined in the current environment.
detailed	a logical specifying if detailed statistics should be provided.
...	additional arguments to be passed to the generic function.
x	an object of class 'describe'.
digits	the number of significant digits to use.

Details

This function provides simple descriptive statistics for different types of variables, such as numeric, factor, and logical, conditioning or not on another variable.

Author(s)

Luca Scrucca

See Also

[summary\(\)](#), [by\(\)](#)

Examples

```
data(warpbreaks)  
describe(warpbreaks)  
describe(warpbreaks, detail = TRUE)  
describe(warpbreaks, by = wool)
```

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

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)

Source

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

See Also

Other datasets in qcc package: [RyanMultivar](#), [antifreeze](#), [boiler](#), [circuit](#), [orangejuice](#), [pcmanufact](#), [pistonrings](#), [viscosity](#)

ellipseChart

*Multivariate Quality Control Charts***Description**

Plot an ellipse chart for a bivariate quality control data.

Usage

```
ellipseChart(
  object,
  chart.all = getOption("qcc.chart.all"),
  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 character string specifying the main title. Set title = FALSE or title = NA to remove the 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 <code>points()</code> 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. (2013). *Introduction to Statistical Quality Control*, 7th ed. New York: John Wiley & Sons.
- Ryan, T. P. (2011). *Statistical Methods for Quality Improvement*, 3rd 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,
  newdata,
  newsizes,
  ...
)

## S3 method for class 'ewma.qcc'
print(x, digits = getOption("digits"), ...)

## S3 method for class 'ewma.qcc'
summary(object, ...)

## S3 method for class 'ewma.qcc'
plot(
  x,
  xtime = NULL,
  add.stats = getOption("qcc.add.stats"),
  chart.all = getOption("qcc.chart.all"),
  fill = getOption("qcc.fill"),
  label.center = "CL",
  label.limits = c("LCL", "UCL"),
  title,
  xlab,
  ylab,
```

```

    xlim,
    ylim,
    digits = getOption("digits"),
    ...
)

```

Arguments

<code>data</code>	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".
<code>sizes</code>	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.
<code>center</code>	a value specifying the center of group statistics or target.
<code>std.dev</code>	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 <code>sd.xbar()</code> and <code>sd.xbar.one()</code> for, respectively, the grouped data case and the individual observations case.
<code>lambda</code>	the smoothing parameter $0 \leq \lambda \leq 1$.
<code>nsigmas</code>	a numeric value specifying the number of sigmas to use for computing control limits.
<code>newdata</code>	a data frame, matrix or vector, as for the data argument, providing further data to plot but not included in the computations.
<code>newsizes</code>	a vector as for the sizes argument providing further data sizes to plot but not included in the computations.
<code>...</code>	additional arguments to be passed to the generic function.
<code>x</code>	an object of class 'ewma.qcc'.
<code>digits</code>	the number of significant digits to use.
<code>object</code>	an object of class 'ewma.qcc'.
<code>xtime</code>	a vector of date-time values as returned by <code>Sys.time()</code> and <code>Sys.Date()</code> . If provided it is used for x-axis so it must be of the same length as the statistic charted.
<code>add.stats</code>	a logical value indicating whether statistics and other information should be printed at the bottom of the chart.
<code>chart.all</code>	a logical value indicating whether both statistics for data and for newdata (if given) should be plotted.
<code>fill</code>	a logical value specifying if the in-control area should be filled with the color specified in the <code>qcc.zones</code> option.
<code>label.center</code>	a character specifying the label for center line.
<code>label.limits</code>	a character vector specifying the labels for control limits.
<code>title</code>	a character string specifying the main title. Set <code>title = NULL</code> to remove the title.

xlab	a string giving the label for the x-axis.
ylab	a string giving the label for the y-axis.
xlim	a numeric vector specifying the limits for the x-axis.
ylim	a numeric vector specifying the limits for the y-axis.

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. (2013). *Introduction to Statistical Quality Control*, 7th ed. New York: John Wiley & Sons.
- Ryan, T. P. (2011). *Statistical Methods for Quality Improvement*, 3rd 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

[ewmaSmooth\(\)](#)

Other control charts: [cusum\(\)](#), [qcc\(\)](#)

Examples

```
## Grouped-data
diameter <- qccGroups(data = pistonrings, diameter, sample)

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

ewma(diameter[1:25,], lambda=0.2, nsigmas=2.7, newdata=diameter[26:40,])

## Individual observations
```

```
q <- with(viscosity, ewma(viscosity[trial], lambda = 0.2, nsigmas = 2.7,
                          newdata = viscosity[!trial]))
summary(q)
plot(q)
```

ewmaSmooth	<i>EWMA smoothing function</i>
------------	--------------------------------

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 $0 \leq \lambda \leq 1$ 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. (2013). *Introduction to Statistical Quality Control*, 7th 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,
  plot = TRUE,
  ...
)

## S3 method for class 'mqcc'
print(x, digits = getOption("digits"), ...)

## S3 method for class 'mqcc'
summary(object, ...)
```

```
## S3 method for class 'mqcc'
plot(
  x,
  add.stats = getOption("qcc.add.stats"),
  chart.all = getOption("qcc.chart.all"),
  fill = getOption("qcc.fill"),
  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: "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 <code>limits.T2()</code> or <code>limits.T2.single()</code>) 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 <code>limits.T2()</code> or <code>limits.T2.single()</code>) 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.

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

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. (2013). *Introduction to Statistical Quality Control*, 7th ed. New York: John Wiley & Sons.
- Ryan, T. P. (2011). *Statistical Methods for Quality Improvement*, 3rd 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
q = mqcc(RyanMultivar, type = "T2")
summary(q)
ellipseChart(q)
ellipseChart(q, show.id = TRUE)
q = mqcc(RyanMultivar, type = "T2", pred.limits = TRUE)

# Xbar-charts for single variables computed adjusting the
# confidence level of the T^2 chart:
q1 = with(RyanMultivar,
           qcc(X1, type = "xbar", confidence.level = q$confidence.level^(1/2)))
summary(q1)
q2 = with(RyanMultivar,
           qcc(X2, type = "xbar", confidence.level = q$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(RyanMultivar, 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(RyanMultivar, type = "T2", newdata = Xnew, pred.limits = TRUE)
summary(qq)

## Individual observations data
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
rob = cov.rob(boiler)
qrob = mqcc(boiler, type = "T2.single", center = rob$center, cov = rob$cov)
summary(qrob)
```

nclass.hist

Histogram number of classes/bins

Description

Computes the optimal number of classes/bins for an histogram as the maximum between the Sturges and Freedman-Diaconis (FD) estimators. For small datasets the Sturges value will usually be chosen, while larger datasets will usually default to FD. Avoids the overly conservative behaviour of FD and Sturges for small and large datasets, respectively. This is the default option in `numpy.histogram_bin_edges` available in Python.

Usage

```
nclass.hist(x, ...)
```

Arguments

`x` a vector of data values.
`...` additional arguments to be passed to low level functions.

Value

The value of suggested number of classes/bins.

Author(s)

Luca Scrucca

See Also

[grDevices::nclass.FD\(\)](#), [grDevices::nclass.Sturges\(\)](#)

Examples

```
set.seed(1)
x <- stats::rnorm(111)
nclass.hist(x)
x <- stats::rnorm(1111)
nclass.hist(x)
```

ocCurves

Operating Characteristic Function

Description

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

Usage

```
ocCurves(object, ...)
```

```
ocCurves.xbar(  
  object,  
  size = c(1, 5, 10, 15, 20),  
  shift = seq(0, 5, by = 0.1),  
  nsigmas = object$nsigmas,  
  ...  
)
```

```
ocCurves.R(  
  object,  
  size = c(2, 5, 10, 15, 20),  
  multiplier = seq(1, 6, by = 0.1),  
  nsigmas = object$nsigmas,  
  ...  
)
```

```
ocCurves.S(  
  object,  
  size = c(2, 5, 10, 15, 20),  
  multiplier = seq(1, 6, by = 0.1),  
  nsigmas = object$nsigmas,  
  ...  
)
```

```
ocCurves.p(object, ...)
```

```
ocCurves.c(object, ...)
```

```
## S3 method for class 'ocCurves'
print(x, digits = getOption("digits"), ...)

## S3 method for class 'ocCurves'
plot(x, what = c("beta", "ARL"), title, xlab, ylab, lty, lwd, col, ...)
```

Arguments

<code>object</code>	an object of class 'qcc'.
<code>...</code>	catches further ignored arguments.
<code>size</code>	a vector of values specifying the sample sizes for which to draw the OC curves.
<code>shift, multiplier</code>	a vector of values specifying the shift or multiplier values (in units of sigma).
<code>nsigmas</code>	a numeric value specifying the number of sigmas to use for computing control limits; if <code>nsigmas</code> is NULL, <code>object\$conf</code> is used to set up probability limits.
<code>x</code>	an object of class 'ocCurves'.
<code>digits</code>	the number of significant digits to use.
<code>what</code>	a string specifying the quantity to plot on the y-axis. Possible values are "beta" for the probability of not detecting a shift, and "ARL" for the average run length.
<code>title</code>	a character string specifying the main title. Set <code>title = NULL</code> to remove the title.
<code>xlab, ylab</code>	a string giving the label for the x-axis and the y-axis.
<code>lty, lwd, col</code>	values or vector of values controlling the line type, line width and colour of curves.

Details

An operating characteristic curve graphically provides information about the probability of not detecting a shift in the process. `ocCurves` 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 returns an object of class 'ocCurves' which contains a matrix or a vector of beta values (the probability of type II error) and ARL (average run length).

Author(s)

Luca Scrucca

References

- Mason, R. L., and Young, J. C. (2002). *Multivariate Statistical Process Control with Industrial Applications*. SIAM.
- Montgomery, D.C. (2013). *Introduction to Statistical Quality Control*, 7th ed. New York: John Wiley & Sons.
- Ryan, T. P. (2011). *Statistical Methods for Quality Improvement*, 3rd 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)
diameter = qccGroups(diameter, sample, data = pistonrings)
oc = ocCurves.xbar(qcc(diameter, type="xbar", nsigmas=3))
oc
plot(oc)

data(orangejuice)
oc = with(orangejuice,
          ocCurves(qcc(D[trial], sizes=size[trial], type="p")))
oc
plot(oc)

data(circuit)
oc = with(circuit,
          ocCurves(qcc(x[trial], sizes=size[trial], type="c")))
oc
plot(oc)
```

orangejuice

Orange juice data

Description

Frozen orange juice concentrate is packaged in 6-ounce cardboard cans. A machine forms each can by spinning it from cardboard stock and attaching a metal bottom panel. A can is classified as nonconforming if an inspection indicates that it could leak when filled, either along the side seam or around the bottom joint.

Usage

orangejuice

orangejuice2

Format

orangejuice is a data frame with 54 observations and orangejuice2 is a data frame with 64 observations. Both contain the following 4 variables:

sample sample id
D number of defectives
size sample sizes
trial trial samples (TRUE/FALSE)

Details

Thirty samples of 50 cans each were collected at half-hour intervals while the machine operated continuously over three shifts. A new batch of cardboard stock was introduced beginning with sample 15, and an inexperienced operator was temporarily assigned to the machine for sample 23. After these 30 samples, the machine was adjusted and 24 additional samples were collected. The orangejuice2 dataset contains samples collected after this adjustment.

Source

Montgomery, D.C. (1991) *Introduction to Statistical Quality Control*, 2nd ed, New York, John Wiley & Sons, pp. 152–159.

See Also

Other datasets in qcc package: [RyanMultivar](#), [antifreeze](#), [boiler](#), [circuit](#), [dyedcloth](#), [pcmanufact](#), [pistonrings](#), [viscosity](#)

paretoChart	<i>Pareto chart</i>
-------------	---------------------

Description

Computes a table of statistics and plot a Pareto chart.

Usage

```
paretoChart(data, ...)

## S3 method for class 'paretoChart'
print(x, digits = getOption("digits") - 3, ...)

## S3 method for class 'paretoChart'
plot(
  x,
  title,
  xlab,
  ylab = "Frequency",
  ylab2 = "Cumulative percentage",
  ylim,
  col = blues.colors(nlevels),
  ...
)
```

Arguments

<code>data</code>	a vector of values. <code>names(data)</code> are used for labelling the bars.
<code>...</code>	catch other optional arguments.
<code>x</code>	an object of class <code>'paretoChart'</code> returned by a call to <code>paretoChart()</code> function.
<code>digits</code>	the number of significant digits to use.
<code>title</code>	a character string specifying the main title. Set <code>title = NULL</code> to remove the title.
<code>xlab</code>	a string specifying the label for the x-axis.
<code>ylab</code>	a string specifying the label for the y-axis.
<code>ylab2</code>	a string specifying the label for the second y-axis on the right side.
<code>ylim</code>	a numeric vector specifying the limits for the y-axis.
<code>col</code>	a value for the color, a vector of colors, or a palette for the bars. See the help for colors() and palette() .
<code>plot</code>	a logical specifying if the chart should be provided (<code>TRUE</code> , default).

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 `'paretoChart'` containing the descriptive statistics used to draw the Pareto chart. This object has associated a `print` and `plot` method.

Author(s)

Luca Scrucca

References

- Mason, R. L., and Young, J. C. (2002). *Multivariate Statistical Process Control with Industrial Applications*. SIAM.
- Montgomery, D.C. (2013). *Introduction to Statistical Quality Control*, 7th ed. New York: John Wiley & Sons.
- Ryan, T. P. (2011). *Statistical Methods for Quality Improvement*, 3rd 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.")
pc = paretoChart(defect, ylab = "Error frequency")
pc
plot(pc)

plot(paretoChart(defect, ylab = "Error frequency"), col=rainbow(length(defect)))
```

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

```
pcmanufact
```

Format

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

x number of nonconformities (inspection units)

size number of computers inspected

Source

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

See Also

Other datasets in qcc package: [RyanMultivar](#), [antifreeze](#), [boiler](#), [circuit](#), [dyedcloth](#), [orangejuice](#), [pistonrings](#), [viscosity](#)

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

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)

Source

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

See Also

Other datasets in qcc package: [RyanMultivar](#), [antifreeze](#), [boiler](#), [circuit](#), [dyedcloth](#), [orangejuice](#), [pcmanufact](#), [viscosity](#)

processCapability	<i>Process capability analysis</i>
-------------------	------------------------------------

Description

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

Usage

```
processCapability(
  object,
  spec.limits,
  target,
  std.dev,
  nsigmas,
  confidence.level = 0.95,
  ...
)

## S3 method for class 'processCapability'
print(x, digits = getOption("digits"), ...)

## S3 method for class 'processCapability'
summary(object, ...)

## S3 method for class 'processCapability'
plot(
  x,
  add.stats = getOption("qcc.add.stats"),
  breaks = nclass.hist,
  fill = adjustcolor(getOption("qcc.zones")$fill, alpha.f = 0.5),
  color = "white",
  title,
  xlab,
  digits = getOption("digits"),
  ...
)
```

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 between 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.
...	catches further ignored arguments.
x	an object of class 'processCapability'.
digits	the number of significant digits to use.
add.stats	a logical value indicating whether statistics and capability indices should be added at the bottom of the chart.
breaks	a value or a function used to select the number of bins in a histogram. See the help for nclass.scott() for more details.
fill, color	values specifying the colour of the filled area and the border used for drawing the histogram.
title	a character string specifying the plot title. Set title = NULL to remove the title.
xlab	a character string specifying the label for the x-axis.

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. (2013). *Introduction to Statistical Quality Control*, 7th 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)
diameter = qccGroups(data = pistonrings, diameter, sample)
q = qcc(diameter[1:25,], type="xbar", nsigmas=3)
pc = processCapability(q, spec.limits=c(73.95,74.05))
pc
plot(pc)
plot(processCapability(q, spec.limits=c(73.95,74.05), target=74.02))
plot(processCapability(q, spec.limits=c(73.99,74.01)))
plot(processCapability(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 = c("xbar", "R", "S", "xbar.one", "p", "np", "c", "u", "g"),
  sizes,
  center,
  std.dev,
  limits,
  newdata,
```

```

    newsizes,
    nsigmas = 3,
    confidence.level,
    rules = c(1, 4),
    rule.set = c("western-electric", "nelson"),
    ...
)

## S3 method for class 'qcc'
print(x, digits = getOption("digits"), ...)

## S3 method for class 'qcc'
summary(object, ...)

## S3 method for class 'qcc'
plot(
  x,
  xtime = NULL,
  add.stats = getOption("qcc.add.stats"),
  chart.all = getOption("qcc.chart.all"),
  fill = getOption("qcc.fill"),
  label.center = "CL",
  label.limits = c("LCL ", "UCL"),
  title,
  xlab,
  ylab,
  xlim,
  ylim,
  digits = getOption("digits"),
  ...
)

```

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:

Type	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 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 in case of a continuous process variable; see sd.xbar() , sd.xbar.one() , sd.R() , and sd.S() .
limits		a two-values vector specifying control limits.
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.
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 value or a vector of values specifying the rules to apply to the chart. See qccRules() for possible values and their meaning.
rule.set		a character string specifying how numeric rules values are interpreted. The default is "western-electric" specifying Western Electric rules 1 through 4. Use "nelson" to apply Nelson rules 1 through 8.
...		additional arguments to be passed to the generic function.
x		an object of class 'qcc'.
digits		the number of significant digits to use.
object		an object of class 'qcc'.
xtime		a vector of date-time values as returned by Sys.time() and Sys.Date() . If provided it is used for x-axis so it must be of the same length as the statistic charted.
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.
fill		a logical value specifying if the in-control area should be filled with the color specified in the qcc.zones option.

<code>label.center</code>	a character specifying the label for center line.
<code>label.limits</code>	a character vector specifying the labels for control limits.
<code>title</code>	a character string specifying the main title. Set <code>title = NULL</code> to remove the title.
<code>xlab</code>	a string giving the label for the x-axis.
<code>ylab</code>	a string giving the label for the y-axis.
<code>xlim</code>	a numeric vector specifying the limits for the x-axis.
<code>ylim</code>	a numeric vector specifying the limits for the y-axis.

Details

Numeric rules values are interpreted within `rule.set`. By default, `rules = c(1, 4)` applies Western Electric rules 1 and 4 for backward compatibility. Nelson rules can be requested with `rules = 1:8`, `rule.set = "nelson"`.

Value

Returns an object of class `'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. (2013). *Introduction to Statistical Quality Control*, 7th ed. New York: John Wiley & Sons.
- Ryan, T. P. (2011). *Statistical Methods for Quality Improvement*, 3rd 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

[qccRules\(\)](#), [ocCurves\(\)](#), [processCapability\(\)](#), [qccGroups\(\)](#)
Other control charts: [cusum\(\)](#), [ewma\(\)](#)

Examples

```
## Continuous data
diameter = qccGroups(data = pistonrings, diameter, sample)

(q = qcc(diameter[1:25,], type="xbar"))
plot(q)
```

```

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

q = qcc(diameter[1:25,], type="xbar", newdata=diameter[26:40,])
plot(q, chart.all=FALSE)

plot(qcc(diameter[1:25,], type="xbar", newdata=diameter[26:40,], nsigmas=2))

plot(qcc(diameter[1:25,], type="xbar", newdata=diameter[26:40,], confidence.level=0.99))

q <- qcc(diameter[1:25,], type="R")
q
plot(q)

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

plot(qcc(diameter[1:25,], type="S"))

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

plot(qcc(diameter[1:25,], type="xbar", newdata=diameter[26:40,], rules = 1:4))

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

# to customize a Shewhart chart use
q <- qcc(diameter[1:25, ], type = "xbar")
graph = plot(q, ylim = c(73.9, 74.1))
# returned object is of class "patchwork", then add geom_* layer to the first element
graph[[1]] <- graph[[1]] +
  geom_hline(yintercept = c(73.95, 74.05), lty = 2)
graph

## Attribute data
q <- with(orangejuice, qcc(D[trial], sizes=size[trial], type="p"))
q
plot(q)

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

```

```

plot(q1)

data(orangejuice2)
q2 <- with(orangejuice2,
           qcc(D[trial], sizes=size[trial], type="p"))
plot(q2)
q2 <- with(orangejuice2,
           qcc(D[trial], sizes=size[trial], type="p",
               newdata=D[!trial], newsize=size[!trial]))
plot(q2)

data(circuit)
plot(with(circuit, qcc(x[trial], sizes=size[trial], type="c"))))

# remove out-of-control points (see help(circuit) for the reasons)
outofctrl = c(15,23)
q1 <- with(orangejuice[-outofctrl,],
           qcc(D[trial], sizes=size[trial], type="p"))
plot(q1)
q1 <- with(orangejuice[-outofctrl,],
           qcc(D[trial], sizes=size[trial], type="p",
               newdata=D[!trial], newsize=size[!trial]))
plot(q1)

outofctrl = c(6,20)
q1 <- with(circuit[-outofctrl,],
           qcc(x[trial], sizes=size[trial], type="c"))
plot(q1)
q1 <- with(circuit[-outofctrl,],
           qcc(x[trial], sizes=size[trial], type="c",
               newdata = x[!trial], newsize = size[!trial]))
plot(q1)
q1 <- with(circuit[-outofctrl,],
           qcc(x[trial], sizes=size[trial], type="u",
               newdata = x[!trial], newsize = size[!trial]))
plot(q1)

data(pcmanufact)
q1 <- with(pcmanufact, qcc(x, sizes=size, type="u"))
q1
plot(q1)

data(dyedcloth)
# variable control limits
plot(with(dyedcloth, qcc(x, sizes=size, type="u"))))
# standardized control chart
q <- with(dyedcloth, qcc(x, sizes=size, type="u"))
z <- (q$statistics - q$center)/sqrt(q$center/q$size)
plot(qcc(z, sizes = 1, type = "u", center = 0, std.dev = 1, limits = c(-3,3)),
     title = "Standardized u chart")

## Continuous one-at-time data

```

```

q <- with(viscosity, qcc(viscosity[trial], type = "xbar.one"))
q
plot(q)
# batch 4 is out-of-control because of a process temperature controller
# failure; remove it and recompute
viscosity <- viscosity[-4,]
plot(with(viscosity,
          qcc(viscosity[trial], type = "xbar.one", newdata = viscosity[!trial])))

```

qcc-deprecated

Deprecated Functions in qcc package

Description

These functions are provided for compatibility with older versions of the package qcc, and they will likely be removed in the future.

Usage

```

pareto.chart(...)

process.capability(...)

oc.curves(...)

qcc.overdispersion.test(...)

cause.and.effect(...)

qcc.groups(...)

```

Arguments

... all arguments are passed down.

qcc.options

qcc Options Interface

Description

qcc.options() is retained as a deprecated compatibility interface for the standard qcc.* options.

Usage

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

Arguments

... No arguments to view all `qcc.*` options; a single legacy option name to retrieve its value; or named values or a named list to update standard options.

Details

Set plotting defaults with `options()` using these names: `qcc.add.stats`, `qcc.chart.all`, `qcc.fill`, `qcc.rules`, `qcc.zones`, `qcc.bg.margin`, `qcc.bg.figure`, `qcc.cex`, `qcc.font.stats`, and `qcc.cex.stats`.

Value

A named list for a call without arguments, a single option value for a character-name lookup, or the updated named list invisibly after a setter call.

qccGroups	<i>Grouping data based on a sample indicator</i>
-----------	--

Description

This function allows to easily group data to use as input to the `qcc()` function.

Usage

```
qccGroups(data, x, sample)
```

Arguments

data	a data frame (or a similar structure which can be coerced to be a <code>data.frame</code>) providing the observed data. If not provided, the input to the following arguments must be objects defined in the calling environment.
x	a name from data or a vector of observed data values.
sample	a name from data or a vector of sample indicators defining the rationale sub-groups of data values.

Value

The function returns a matrix of suitable dimensions. If one or more group have fewer observations than others, NA values are used to fill empty values.

Author(s)

Luca Scrucca

See Also

[qcc\(\)](#)

Examples

```
data(pistonrings)
# create a matrix of 40 samples made of 5 observations each
qccGroups(data = pistonrings, diameter, sample)
# remove some observations to get still a 40x5 matrix but filled with NAs
qccGroups(data = pistonrings[-c(1,2,50,52,199),], diameter, sample)
```

qccOverdispersionTest *Overdispersion test for binomial and poisson data*

Description

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

Usage

```
qccOverdispersionTest(
  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 distribution is assumed, otherwise a poisson distribution.

Details

This very simple test amounts to compute the test statistic

$$D = s^2 / \sigma^2 \times (n - 1)$$

where s^2 is the observed variance, σ^2 is the theoretical variance, and n is the number of observations. The test statistic is then compared to the critical value of a Chi-square distribution with $n - 1$ 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 & Hall.

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))
qccOverdispersionTest(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)
qccOverdispersionTest(x)
```

rules	<i>Rules for Shewhart charts</i>
-------	----------------------------------

Description

Functions which implement rules to signal out-of-control (OOC) or out-of-trend (OOT) points in Shewhart control charts.

Usage

```
qccRules(object, rules = object$rules, rule.set = object$rule.set)

qccRulesViolatingWER1(object, limits = object$limits)

qccRulesViolatingWER2(
  object,
  run.points = 2,
  run.length = 3,
  k = object$nsigmas * 2/3
)

qccRulesViolatingWER3(object, ...)

qccRulesViolatingWER4(object)

qccRulesViolatingNEL1(object)

qccRulesViolatingNEL2(object, run.length = 9)

qccRulesViolatingNEL3(object)
```

```
qccRulesViolatingNEL4(object)
```

```
qccRulesViolatingNEL5(object)
```

```
qccRulesViolatingNEL6(object)
```

```
qccRulesViolatingNEL7(object)
```

```
qccRulesViolatingNEL8(object)
```

Arguments

<code>object</code>	an object of class 'qcc'.
<code>rules</code>	a vector of values specifying the rules used to declare a process OOC or OOT. Numeric values are interpreted differently depending on the value of <code>rule.set</code> .
<code>rule.set</code>	a character string selecting the rule set used to interpret numeric rules values. Use "western-electric" for Western Electric rules or "nelson" for Nelson rules. default: "western-electric"
<code>limits</code>	a two-column matrix containing the lower and upper control limits.
<code>run.points</code>	the minimum number of points beyond the sigma limit needed to signal a violation.
<code>run.length</code>	the number of consecutive points in the run.
<code>k</code>	the number of standard deviations from the center line used to define the sigma limits.
<code>...</code>	additional arguments, currently ignored.

Details

The `qccRulesViolatingWER1`, `qccRulesViolatingWER2`, ..., `qccRulesViolatingWER4` functions return the indices violating the corresponding Western Electric rule:

1. One point plots outside 3-sigma control limits.
2. Two of three consecutive points plot beyond a 2-sigma limit.
3. Four of five consecutive points plot beyond a 1-sigma limit.
4. Eight consecutive points plot on one side of the center line.

The `qccRulesViolatingNEL1`, `qccRulesViolatingNEL2`, ..., `qccRulesViolatingNEL8` functions return the indices violating the corresponding Nelson rule:

1. One point plots outside 3-sigma control limits.
2. Nine points in a row plot on the same side of the center line.
3. Six points in a row are steadily increasing or decreasing.
4. Fourteen points in a row alternate up and down.
5. Two of three consecutive points plot beyond a 2-sigma limit on the same side.
6. Four of five consecutive points plot beyond a 1-sigma limit on the same side.
7. Fifteen points in a row plot within 1 sigma of the center line.
8. Eight points in a row plot outside 1 sigma on both sides of the center line.

Value

The function `qccRules` returns a vector of the same length as the statistics charted with value NA if no rule is violated, and numerical values corresponding to the violated rule. If more than one rule is violated, the value returned corresponds to the most serious violation.

Author(s)

Luca Scrucca

Anas Sheashaey

RyanMultivar

Ryan's (2011) multivariate data for quality control

Description

Multivariate data from Ryan (2011, Table 9.2).

Usage

RyanMultivar

Format

Multivariate data on 20 samples of size 4 for two variables:

RyanMultivar a list of two data frames, X1 and X2, one for each variable.

Source

Ryan, T. P. (2011), *Statistical Methods for Quality Improvement*, 3rd ed. New York: John Wiley & Sons, Inc.

See Also

Other datasets in qcc package: [antifreeze](#), [boiler](#), [circuit](#), [dyedcloth](#), [orangejuice](#), [pcmanufact](#), [pistonrings](#), [viscosity](#)

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, nsigmas = NULL, conf = NULL)
```

Arguments

data	The observed data values.
sizes	The sample sizes.
...	catches further ignored arguments.
center	The sample/group center statistic.
std.dev	The within-group standard deviation.
nsigmas	Number of sigmas used to compute control limits. Ignored when the conf argument is provided.
conf	Confidence level used to compute control limits. Must be a numeric value in (0, 1).

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. (2013). *Introduction to Statistical Quality Control*, 7th 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, nsigmas = NULL, conf = NULL)
```

Arguments

data	The observed data values.
sizes	sample sizes (not used)
...	catches further ignored arguments.
center	The sample/group center statistic.
std.dev	standard deviation of geometric distribution
nsigmas	Number of sigmas used to compute control limits. Ignored when the conf argument is provided.
conf	Confidence level used to compute control limits. Must be a numeric value in (0, 1).

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.

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

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.

Author(s)

Greg Snow (greg.snow@ihc.com)

References

- Kaminsky, F. C., et al. (1992). *Statistical Control Charts Based on a Geometric Distribution*. *Journal of Quality Technology*, **24**, 63–69.
- Yang, Z., et al. (2002). On the Performance of Geometric Charts with Estimated Control Limits. *Journal of Quality Technology*, **34**, 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, nsigmas = NULL, conf = NULL)
```

Arguments

data	The observed data values.
sizes	The sample sizes.
...	catches further ignored arguments.
center	The sample/group center statistic.
std.dev	The within-group standard deviation.
nsigmas	Number of sigmas used to compute control limits. Ignored when the conf argument is provided.
conf	Confidence level used to compute control limits. Must be a numeric value in (0, 1).

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. (2013). *Introduction to Statistical Quality Control*, 7th 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, nsigmas = NULL, conf = NULL)
```

Arguments

<code>data</code>	The observed data values.
<code>sizes</code>	The sample sizes.
<code>...</code>	catches further ignored arguments.
<code>center</code>	The sample/group center statistic.
<code>std.dev</code>	The within-group standard deviation.
<code>nsigmas</code>	Number of sigmas used to compute control limits. Ignored when the <code>conf</code> argument is provided.
<code>conf</code>	Confidence level used to compute control limits. Must be a numeric value in (0, 1).

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. (2013). *Introduction to Statistical Quality Control*, 7th 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, nsigmas = NULL, conf = NULL)
```

Arguments

<code>data</code>	The observed data values.
<code>sizes</code>	samples sizes. Optional
<code>std.dev</code>	within group standard deviation. Optional for <code>sd.R</code> function, required for <code>limits.R</code> . See sd.xbar() .
<code>...</code>	catches further ignored arguments.
<code>center</code>	The sample/group center statistic.
<code>nsigmas</code>	Number of sigmas used to compute control limits. Ignored when the <code>conf</code> argument is provided.
<code>conf</code>	Confidence level used to compute control limits. Must be a numeric value in (0, 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. (2013). *Introduction to Statistical Quality Control*, 7th 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, nsigmas = NULL, conf = NULL)
```

Arguments

<code>data</code>	The observed data values.
<code>sizes</code>	samples sizes. Optional
<code>std.dev</code>	within group standard deviation. Optional for <code>sd.S</code> function, required for <code>limits.S</code> . See sd.xbar() .
<code>...</code>	catches further ignored arguments.
<code>center</code>	The sample/group center statistic.
<code>nsigmas</code>	Number of sigmas used to compute control limits. Ignored when the <code>conf</code> argument is provided.
<code>conf</code>	Confidence level used to compute control limits. Must be a numeric value in (0, 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. (2013). *Introduction to Statistical Quality Control*, 7th 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	<i>Statistics used in computing and drawing the Hotelling T^2 chart for subgrouped data</i>
----------	--

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

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

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`: prediction 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. (2013). *Introduction to Statistical Quality Control*, 7th ed. New York: John Wiley & Sons.
- Ryan, T. P. (2011). *Statistical Methods for Quality Improvement*, 3rd ed. New York: John Wiley & Sons, Inc.

See Also

`mqcc()`, `stats.T2.single()`

<code>stats.T2.single</code>	<i>Statistics used in computing and drawing the Hotelling T^2 chart for individual observations data</i>
------------------------------	---

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 = 1, 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 ($0 < \text{conf} < 1$)

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, equal to data since samples are of size one.
- **center**: sample/group center statistic.
- **S**: covariance matrix.

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

- **control**: control limits.
- **prediction**: prediction 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. (2013). *Introduction to Statistical Quality Control*, 7th ed. New York: John Wiley & Sons.
- Ryan, T. P. (2011). *Statistical Methods for Quality Improvement*, 3rd 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, nsigmas = NULL, conf = NULL)
```

Arguments

data	The observed data values.
sizes	The sample sizes.
...	catches further ignored arguments.
center	The sample/group center statistic.
std.dev	The within-group standard deviation.
nsigmas	Number of sigmas used to compute control limits. Ignored when the conf argument is provided.
conf	Confidence level used to compute control limits. Must be a numeric value in (0, 1).

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. (2013). *Introduction to Statistical Quality Control*, 7th 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	<i>Statistics used in computing and drawing a Shewhart xbar chart</i>
------------	---

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, nsigmas = NULL, conf = NULL)
```

Arguments

data	The observed data values.
sizes	samples sizes. Optional
std.dev	within group standard deviation. Optional for sd.xbar function, required for limits.xbar. See details.
...	catches further ignored arguments.
center	The sample/group center statistic.
nsigmas	Number of sigmas used to compute control limits. Ignored when the conf argument is provided.
conf	Confidence level used to compute control limits. Must be a numeric value in (0, 1).

Details

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

- "UWAVE-R": UnWeighted AVErage of within-group estimates based on within-group Ranges.
- "UWAVE-SD": UnWeighted AVErage of within-group estimates based on within-group Standard Deviations.
- "MVLUE-R": Minimum Variance Linear Unbiased Estimator computed as a weighted average of within-group estimates based on within-group Ranges.

- "MVLUE-SD": Minimum Variance Linear Unbiased Estimator computed as a weighted average of within-group estimates based on within-group Standard Deviations.
- "RMSDF": Root-Mean-Square estimator computed as a weighted average of within-group estimates based on within-group 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"	available	not available	default
"MVLUE-R"	available	available	not available
"MVLUE-SD"	available	not available	available
"RMSDF"	available	not available	available

Detailed definitions of formulae implemented are available in the SAS/QC 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. (2013). *Introduction to Statistical Quality Control*, 7th 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	<i>Statistics used in computing and drawing a Shewhart xbar chart for one-at-time data</i>
----------------	--

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", "MSSD"), r = 2, ...)
```

```
limits.xbar.one(center, std.dev, sizes, nsigmas = NULL, conf = NULL)
```

Arguments

data	The observed data values.
sizes	samples sizes. Not needed, size = 1 is used.
std.dev	within group standard deviation. Optional for sd.xbar.one function, required for limits.xbar.one. See details.
r	number of successive observations in each moving range for the "MR" estimator. Ignored for "SD" and "MSSD".
...	catches further ignored arguments.
center	The sample/group center statistic.
nsigmas	Number of sigmas used to compute control limits. Ignored when the conf argument is provided.
conf	Confidence level used to compute control limits. Must be a numeric value in (0, 1).

Details

Methods available for estimating the process standard deviation:

- "MR": moving range; this estimator is based on the scaled mean of moving ranges.
- "SD": sample standard deviation; this estimator is defined as $\text{sd}(x) / c_4(n)$, where n is the number of individual measurements of x .
- "MSSD": mean squared successive differences; this estimator is defined as $\sqrt{\frac{1}{2(n-1)} \sum_{i=1}^{n-1} (x_{i+1} - x_i)^2} / c'_4(n)$. The correction factor $c'_4(n)$ differs from $c_4(n)$ by accounting for the dependence between successive differences.

For independent normal observations from an in-control process, "MSSD" generally has lower sampling variance than "MR" with $r = 2$. It can also perform well under some changes in the process mean. However, squaring the differences makes it sensitive to outliers. Serial correlation in the observations can also bias the estimate; the correction factor accounts for dependence between differences, not between observations. See Braun and Park (2008) for comparisons of estimators for individuals charts.

Missing values are omitted when estimating the standard deviation. For "MR" and "MSSD", successive observations refer to the remaining values in their original order, including pairs across gaps left by missing values.

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. (2013). *Introduction to Statistical Quality Control*, 7th ed. New York: John Wiley & Sons.
- Ryan, T. P. (2011). *Statistical Methods for Quality Improvement*, 3rd ed. New York: John Wiley & Sons, Inc.
- Wetherill, G. B., and Brown, D. W. (1991). *Statistical Process Control*. New York: Chapman & Hall.
- Braun, W. J., and Park, D. (2008). Estimation of sigma for Individuals Charts. *Journal of Quality Technology*, **40**(3), 332–344. <https://doi.org/10.1080/00224065.2008.11917738>.

See Also

[qcc\(\)](#)

Examples

```
x <- antifreeze[["water"]] # See `?antifreeze`
# 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")
# 3) using bias-corrected MSSD
qcc(x, type="xbar.one", std.dev = "MSSD", 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

```

viscosity

*Viscosity for aircraft primer paint data***Description**

"The viscosity of an aircraft primer paint is an important quality characteristic. The product is produced in batches, and because each batch takes several hours to produce, the production rate is too slow to allow for rational subgroups of size greater than one." (Montgomery, 2005, p. 232)

Usage

```
viscosity
```

Format

A data frame with 35 observations on the following 3 variables.

batch batch number

viscosity viscosity measure

trial preliminary sample indicator (TRUE/FALSE)

Source

Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, 5th ed, New York, John Wiley & Sons, pp. 232-235

See Also

Other datasets in qcc package: [RyanMultivar](#), [antifreeze](#), [boiler](#), [circuit](#), [dyedcloth](#), [orangejuice](#), [pcmanufact](#), [pistonrings](#)

Index

- * **constants for Shewhart charts**
 - c4, [5](#)
 - d2, [11](#)
 - d3, [12](#)
- * **control charts**
 - cusum, [7](#)
 - ewma, [16](#)
 - qcc, [34](#)
- * **datasets in**
 - antifreeze, [3](#)
 - boiler, [4](#)
 - circuit, [7](#)
 - dyedcloth, [14](#)
 - orangejuice, [27](#)
 - pcmanufact, [30](#)
 - pistonrings, [31](#)
 - RyanMultivar, [45](#)
 - viscosity, [60](#)
- * **datasets**
 - antifreeze, [3](#)
 - boiler, [4](#)
 - circuit, [7](#)
 - dyedcloth, [14](#)
 - orangejuice, [27](#)
 - pcmanufact, [30](#)
 - pistonrings, [31](#)
 - RyanMultivar, [45](#)
 - viscosity, [60](#)
- * **htest**
 - qccOverdispersionTest, [42](#)
- * **list(qcc)**
 - antifreeze, [3](#)
 - boiler, [4](#)
 - circuit, [7](#)
 - dyedcloth, [14](#)
 - orangejuice, [27](#)
 - pcmanufact, [30](#)
 - pistonrings, [31](#)
 - RyanMultivar, [45](#)
 - viscosity, [60](#)
- * **package**
 - antifreeze, [3](#)
 - boiler, [4](#)
 - circuit, [7](#)
 - dyedcloth, [14](#)
 - orangejuice, [27](#)
 - pcmanufact, [30](#)
 - pistonrings, [31](#)
 - RyanMultivar, [45](#)
 - viscosity, [60](#)
- antifreeze, [3](#), [4](#), [7](#), [14](#), [28](#), [31](#), [45](#), [60](#)
- barplot(), [30](#)
- blues.colors, [3](#)
- boiler, [3](#), [4](#), [7](#), [14](#), [28](#), [31](#), [45](#), [60](#)
- by(), [13](#)
- c4, [5](#)
- c4(), [11](#), [12](#), [58](#)
- cause.and.effect (qcc-deprecated), [40](#)
- causeEffectDiagram, [5](#)
- circuit, [3](#), [4](#), [7](#), [14](#), [28](#), [31](#), [45](#), [60](#)
- colors(), [29](#)
- cusum, [7](#)
- cusum(), [18](#), [20](#), [37](#)
- d2, [11](#)
- d2(), [5](#), [12](#)
- d3, [12](#)
- d3(), [5](#), [11](#)
- describe, [13](#)
- dyedcloth, [3](#), [4](#), [7](#), [14](#), [28](#), [31](#), [45](#), [60](#)
- ellipseChart, [14](#)
- ellipseChart(), [23](#)
- ewma, [16](#)
- ewma(), [10](#), [37](#)
- ewmaSmooth, [19](#)
- ewmaSmooth(), [18](#)

- grDevices::nclass.FD(), 24
- grDevices::nclass.Sturges(), 24
- IQCC::c4(), 5
- IQCC::d2(), 11
- IQCC::d3(), 12
- limits.c(stats.c), 46
- limits.g(stats.g), 47
- limits.np(stats.np), 48
- limits.p(stats.p), 49
- limits.R(stats.R), 50
- limits.S(stats.S), 51
- limits.T2(stats.T2), 52
- limits.T2(), 21, 23
- limits.T2.single(stats.T2.single), 53
- limits.T2.single(), 21, 23
- limits.u(stats.u), 55
- limits.xbar(stats.xbar), 56
- limits.xbar.one(stats.xbar.one), 58
- mqcc, 20
- mqcc(), 16, 53, 54
- nclass.hist, 24
- nclass.scott(), 33
- oc.curves(qcc-deprecated), 40
- ocCurves, 25
- ocCurves(), 37
- options(), 41
- orangejuice, 3, 4, 7, 14, 27, 31, 45, 60
- orangejuice2(orangejuice), 27
- palette(), 29
- pareto.chart(qcc-deprecated), 40
- paretoChart, 28
- pcmanufact, 3, 4, 7, 14, 28, 30, 31, 45, 60
- pistonrings, 3, 4, 7, 14, 28, 31, 31, 45, 60
- plot.cusum.qcc(cusum), 7
- plot.ewma.qcc(ewma), 16
- plot.mqcc(mqcc), 20
- plot.ocCurves(ocCurves), 25
- plot.paretoChart(paretoChart), 28
- plot.processCapability
 - (processCapability), 32
- plot.qcc(qcc), 34
- points(), 15
- print.cusum.qcc(cusum), 7
- print.describe(describe), 13
- print.ewma.qcc(ewma), 16
- print.mqcc(mqcc), 20
- print.ocCurves(ocCurves), 25
- print.paretoChart(paretoChart), 28
- print.processCapability
 - (processCapability), 32
- print.qcc(qcc), 34
- process.capability(qcc-deprecated), 40
- processCapability, 32
- processCapability(), 37
- qcc, 34
- qcc(), 10, 18, 20, 23, 27, 34, 41, 47–52, 56, 57, 59
- qcc-deprecated, 40
- qcc.groups(qcc-deprecated), 40
- qcc.options, 40
- qcc.overdispersion.test
 - (qcc-deprecated), 40
- qccGroups, 41
- qccGroups(), 37
- qccOverdispersionTest, 42
- qccRules(rules), 43
- qccRules(), 36, 37
- qccRulesViolatingNEL1(rules), 43
- qccRulesViolatingNEL2(rules), 43
- qccRulesViolatingNEL3(rules), 43
- qccRulesViolatingNEL4(rules), 43
- qccRulesViolatingNEL5(rules), 43
- qccRulesViolatingNEL6(rules), 43
- qccRulesViolatingNEL7(rules), 43
- qccRulesViolatingNEL8(rules), 43
- qccRulesViolatingWER1(rules), 43
- qccRulesViolatingWER2(rules), 43
- qccRulesViolatingWER3(rules), 43
- qccRulesViolatingWER4(rules), 43
- rules, 43
- RyanMultivar, 3, 4, 7, 14, 28, 31, 45, 60
- sd.c(stats.c), 46
- sd.g(stats.g), 47
- sd.np(stats.np), 48
- sd.p(stats.p), 49
- sd.R(stats.R), 50
- sd.R(), 36
- sd.S(stats.S), 51
- sd.S(), 36
- sd.u(stats.u), 55

`sd.xbar(stats.xbar)`, 56
`sd.xbar()`, 9, 17, 36, 50, 51
`sd.xbar.one(stats.xbar.one)`, 58
`sd.xbar.one()`, 9, 17, 36
`shewhartr::shewhart_constants()`, 5, 11, 12
`SixSigma::ss.cc.getc4()`, 5
`SixSigma::ss.cc.getd2()`, 11
`SixSigma::ss.cc.getd3()`, 12
`stats.c`, 46
`stats.g`, 47
`stats.np`, 48
`stats.p`, 49
`stats.R`, 50
`stats.S`, 51
`stats.T2`, 52
`stats.T2()`, 16, 23, 54
`stats.T2.single`, 53
`stats.T2.single()`, 16, 23, 53
`stats.u`, 55
`stats.xbar`, 56
`stats.xbar()`, 36
`stats.xbar.one`, 58
`summary()`, 13
`summary.cusum.qcc(cusum)`, 7
`summary.ewma.qcc(ewma)`, 16
`summary.mqcc(mqcc)`, 20
`summary.processCapability`
 (`processCapability`), 32
`summary.qcc(qcc)`, 34
`Sys.Date()`, 9, 17, 36
`Sys.time()`, 9, 17, 36

viscosity, 3, 4, 7, 14, 28, 31, 45, 60