Package 'bcc'

April 28, 2020

Title Beta Control Charts

Version 1.3.1

Description Applies beta control charts to defined values, using 'qcc' package with new beta control limits. The Beta Chart presents the control limits based on the Beta probability distribution. Can be used for monitoring fraction data from Binomial distribution as replacement of the p-Charts. The Beta Chart was applied for monitoring the variables in three real studies, and it was compared to the control limits with three schemes. The comparative analysis showed that: (i) Beta approximation to the Binomial distribution was more appropriate with values confined in the [0, 1]- interval; and (ii) the charts proposed were more sensitive to the average run length (ARL), in both in-control and out-of-control processes monitoring. The Beta Charts outperform the Shewhart control charts analyzed for monitoring fraction data. Ângelo Márcio Oliveira Sant'Anna, Carla Schwengber ten Caten (2012) <doi:10.1016/j.eswa.2012.02.146>.

Depends R (>= 2.10)

URL https://danielucas28.github.io/

License GPL

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports qcc, methods

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Ângelo Santanna [aut],
Daniel Cerqueira [aut, cre],
Luca Scrucca [ctb]

Maintainer Daniel Cerqueira <danielcerqueira@live.com>

Repository CRAN

Date/Publication 2020-04-28 11:20:11 UTC

R topics documented:

bcc *Beta Control Charts*

Description

Create an object of class 'qcc' to perform statistical quality control. This object may then be used to plot Beta Control Charts.

Usage

```
bcc(data, type = c("1", "2"), sizes, center, std.dev, limits, data.name,
  labels, newdata, newsizes, newdata.name, newlabels, nsigmas = 3,
  confidence.level = 0.9, rules = shewhart.rules, plot = 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. There are two possible types: 1 for discrete data and 2 for continuous data.
`sizes`	a value or a vector of values specifying the sample sizes associated with each group. If data is continuous this parameter should be ignored
`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.
`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 value or a vector of values specifying the rules to apply to the chart. See shewhart.rules for possible values and their meaning.

`plot` logical. If `TRUE` a Shewhart chart is plotted.

`...` further arguments are ignored.

Details

The Beta Chart presents the control limits based on the Beta probability distribution. It was can be used for monitoring fraction data from Binomial distribution as replacement of the p-Charts. The Beta Chart was applied for monitoring the variables in three real studies, and it was compared to the control limits with three schemes. The comparative analysis showed that: (i) Beta approximation to the Binomial distribution was more appropriate with values confined in the [0, 1]- interval; and (ii) the charts proposed were more sensitive to the average run length (ARL), in both in-control and out-of-control processes monitoring. The Beta Charts outperform the Shewhart control charts analyzed for monitoring fraction data.This package was made based on the qcc package. See qcc.

Value

Returns an object of class `'qcc'`.

References

SANT'ANNA, Ângelo M. O; CATEN, Carla Schwengber ten. Beta control charts forsave monitoring fraction data. Expert Systems With Applications, p. 10236-10243. 1 set. 2012.

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

Examples

```
data("Drapper1998data")
bcc(data = Drapper1998data, type = "2")
data("Montgomery2005")
bcc(data=Montgomery2005$Defective, sizes = Montgomery2005$Sample, type=1)
```

`Drapper1998data` *Drapper 1998 data*

Description

Drapper1998data consists of a data set of the study of contaminated peanut by toxic substances in 34 batches of 120 pounds.

Usage

```
data(Drapper1998data)
```

Format

A data frame with 34 observations on the following 1 variable.

PROP a numeric vector

References

Draper, N. R., & Smith, H. (1998) *Applied Regression Analysis*, New York: John Wiley & Sons. 706p.

Examples

```
data(Drapper1998data)
attach(Drapper1998data)
boxplot(Drapper1998data)
plot(Drapper1998data)
```

`limits.beta` *Limits for discrete data*

Description

These function are used to compute the upper and lower control limits.

Usage

```
limits.beta(center, std.dev, sizes, conf)
```

Arguments

`center` sample/group center statistic.

`std.dev` within group standard deviation.

`sizes` sample sizes.

`conf` a numeric value used t o comput control limits, specifying the confidence level (0 < conf < 1)

Value

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

Examples

```
data(Montgomery2005)
limits.beta(center = 0.2313333, std.dev = 0.421685, sizes = Montgomery2005$Sample, conf = 0.9)
```

`limits.beta.p` *Limits for continuos data*

Description

These function are used to compute the upper and lower control limits.

Usage

```
limits.beta.p(center, std.dev, sizes, conf, ...)
```

Arguments

`center`	sample/group center statistic.
`std.dev`	within group standard deviation.
`sizes`	sample sizes.
`conf`	a numeric value used t o comput control limits, specifying the confidence level (if 0 < conf < 1)
`...`	further arguments are ignored.

Value

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

Examples

```
limits.beta.p(center = 0.9989597, std.dev = 0.0009362578, conf = 0.9)
```

`Montgomery2005` *Montgomery 2005 data*

Description

Montgomery2005 consists of a data set of a manufacturing process offrozen orange juice concentrate in 30 packages of 50 units each.

Usage

```
data(Montgomery2005)
```

Format

A data frame with 30 observations on the following 2 variables.

Sample a numeric vector

Defective a numeric vector

References

Montgomery, D. C. (2005). *Introduction to Statistical Quality Control(5th ed.)*, NewYork: John Wiley & Sons. 732p.

Examples

```
data(Montgomery2005)
attach(Montgomery2005)
boxplot(Montgomery2005)
plot(Montgomery2005)
```

`sd.beta` *sd.beta*

Description

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

Usage

```
sd.beta(data, sizes, ...)
```

Arguments

`data`	the observed data values.
`sizes`	sample sizes
`...`	further arguments are ignored.

Details

Performs the calculation of the standard deviation `std.dev` that will be used in the construction of the control chart.

Value

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

Examples

```
data(Montgomery2005)
sd.beta(Montgomery2005$Defective, Montgomery2005$Sample)
```

`sd.beta.p` *sd.beta.p*

Description

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

Usage

```
sd.beta.p(data, sizes, std.dev)
```

Arguments

`data`	the observed data values.
`sizes`	sample sizes
`std.dev`	within group standard deviation.

Details

Performs the calculation of the standard deviation `sd` that will be used in the construction of the control chart.

Value

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

Examples

```
data(Drapper1998data)
sd.beta.p(Drapper1998data)
```

stats.beta *stats.beta*

Description

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

Usage

```
stats.beta(data, sizes)
```

Arguments

`data`	the observed data values.
`sizes`	sample sizes

Details

Provides a list containing the centerline of the chart `pbar` and the data manipulated to be used in the construction of the chart `data/sizes`.

Value

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

Examples

```
data(Montgomery2005)
stats.beta(Montgomery2005$Defective, Montgomery2005$Sample)
```

stats.beta.p *stats.beta.p*

Description

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

Usage

```
stats.beta.p(data,sizes)
```

Arguments

`data`	the observed data values.
`sizes`	sample sizes

Details

Provides a list containing the center line of the graph `center` and the data to be used in the construction of the chart `statistics`.

Value

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

Examples

```
data(Drapper1998data)
sd.beta.p(Drapper1998data)
```

Index