Package 'processcontrol'

June 6, 2025

Type Package

Title Statistical Process Control Charts

Version 0.1.0

Author Peter P. Lupo

Maintainer Peter P. Lupo <pplupo@gmail.com>

Description Generate time series chart for individual values with mean and +/- 3 standard deviation lines and the corresponding mR chart with the upper control limit. Also execute the 8 Shewhart stability run tests and display the violations.

Depends R (>= 4.5.0)

License MIT + file LICENSE

LazyData TRUE

Imports plyr

RoxygenNote 5.0.1

Contents

`six_sigma_ctrl_chart` *Generate IndX and mR charts.*

Description

Generate IndX and mR charts.

Usage

```
six_sigma_ctrl_chart(x, linesColors = c("gray50", "gray65", "gray85", "black",
  "gray85", "gray65", "gray50"), applyRules = c(TRUE, TRUE, TRUE, TRUE, TRUE,
  TRUE, TRUE, TRUE), rulesColors = c("red", "yellow2", "green", "magenta",
  "blue", "orange", "brown", "cyan"), seg = c(), keepStats = TRUE,
  verbose = FALSE)
```

Arguments

x	(mandatory) A data frame with the individual values in the first column and the time in the second column. It can be either a factor, a date or a string and it will be ordered automatically. See `?spcTimeSeries`
linesColors	(optional) A vector with 7 colors in order from the average + 3 standard deviations to the average - 3 standard deviations, including the average itself in the center. Default value is `c("gray50", "gray65", "gray85", "black", "gray85", "gray65", "gray50")`
applyRules	(optional) A vector with 8 boolean values indicating which rules must be applied. Default value is `c(TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE)`
rulesColors	(optional) A vector with colors, one for each rule. The last point of each violating run will be colored indicating the violation and the corresponding rule. Default value is `c("red", "yellow2", "green", "magenta", "blue", "orange", "brown", "cyan")`
seg	(optional) A vector with the positions of the points where there should be breaks and another pair of charts should be plotted. It may be used for better visualization when the series is too long. Default value is `c()`
keepStats	(optional) A boolean indicating if each segment's plot should be considered as part of the same series or independelty, as different series. If TRUE, it will be considered as part of the same series. If FALSE, each plot will have it's limits calculated independently, as well as the application of the rules. It's useful to compare different scenarios. Default value is `TRUE`
verbose	(optional) A boolean indicating if mean, standard deviation/UCL and number of violations should be printed. Default value is `FALSE`

Value

None

References

Engineering Statistics Handbook 6.3.2, NIST/SEMATECH e-Handbook of Statistical Methods National Institute of Standards and Technology, Dec 2006

https://en.wikipedia.org/wiki/Nelson_rules

Lloyd S. Nelson, "Technical Aids," Journal of Quality Technology 16, no. 4 (October 1984), 238-239.

The 8 rules are: 1 One point is more than 3 standard deviations from the mean. 2 Nine (or more) points in a row are on the same side of the mean. 3 Six (or more) points in a row are continually increasing (or decreasing). 4 Fourteen (or more) points in a row alternate in direction, increasing then decreasing. 5 Two (or three) out of three points in a row are more than 2 standard deviations from the mean in the same direction. 6 Four (or five) out of five points in a row are more than 1 standard deviation from the mean in the same direction. 7 Fifteen points in a row are all within 1 standard deviation of the mean on either side of the mean. 8 Eight points in a row exist with none within 1 standard deviation of the mean and the points are in both directions from the mean.

Examples

```
data("spcTimeSeries")
six_sigma_ctrl_chart(spcTimeSeries)
six_sigma_ctrl_chart(spcTimeSeries, verbose=TRUE)
six_sigma_ctrl_chart(spcTimeSeries, seg=c(25, 50, 75))
six_sigma_ctrl_chart(spcTimeSeries, seg=c(25, 50, 75), keepStats=FALSE, verbose=TRUE)
```

---

spcTimeSeries *Time series with one of each violation to be used as test and example sample.*

Description

Time series with one of each violation to be used as test and example sample.

Usage

```
spcTimeSeries
```

Format

A data frame with 100 rows and 2 variables:

values 100 values ranging from 5 to 31

dates 100 dates from 1-jan-2015 to 10-apr-2015 to give the values a time order

Details

It was built with the following code:

codevalues = cbind(values=c(17,
39, # rule 1, point 2 red
11,12,10,5,16,8,5,15,14, # rule 2, point 11 yellow3
21,

```
17,18,19,21,22,25, # rule 3, point 18 green
6,
7,18,9,22,18,21,16,21,18,20,18,24,18,23, # rule 4, point 33 magenta
31,15,31, # rule 5, point 36 blue
10,
25,24,9,24,26, # rule 6, point 42 orange
17,18,17,20,18,11,14,19,15,16,18,13,22,20,10, # rule 7, point 57 brown
7,25,24,8,24,8,25,24, #rule 8, point 65 cyan
22,5,21,16,12,11,33,17,15,13,22,13,11,8,23,5,10,6,10,21,5,9,11,20,8,23,14,19,5,12,17,7,15,14,9))
dates = as.character(seq(from=as.Date("1/1/2015", "%d/%m/%Y"), length.out = 100, by = "days"))

series <- data.frame(values, dates)

series <- series[sample(nrow(series)),]
```

Index