Package: cusumcharter (via r-universe)

August 22, 2024

Title Easier CUSUM Control Charts

Version 0.1.9000

Description Create CUSUM (cumulative sum) statistics from a vector or dataframe. Also create single or faceted CUSUM control charts, with or without control limits. Accepts vector, dataframe, tibble or data.table inputs.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Suggests covr, dplyr, knitr, rmarkdown, testthat (>= 3.0.0), tibble

Config/testthat/edition 3

URL https://github.com/johnmackintosh/cusumcharter, https://johnmackintosh.github.io/cusumcharter/

BugReports https://github.com/johnmackintosh/cusumcharter/issues

Imports rlang, ggplot2, data.table

VignetteBuilder knitr

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

RemoteUrl https://github.com/johnmackintosh/cusumcharter

RemoteRef HEAD

RemoteSha d608750d8499eeeedbf2ee90e933f80a8a0d47c2

Contents

`cusum_control` *Calculate cusum control limits centred on target value or mean*

Description

Calculates cusum statistics and control limits based on input parameters. If no target value is supplied the mean of the x value will be used.

Usage

```
cusum_control(
  x,
  target = NULL,
  std_dev = NULL,
  desired_shift = 1,
  k = 0.5,
  h = 4
)
```

Arguments

`x`	input vector
`target`	target value for comparison, the mean of x will be used if missing
`std_dev`	Defaults to the screened moving range of x. A known or desired value for standard deviation can be supplied instead.
`desired_shift`	how many standard deviations do you want to detect? This value is typically between 0.5 to 1. Defaults to 1.
`k`	allowable slack - defaults to half the standard deviation multiplied by desired shift
`h`	action limits - usually between 4 and 5, defaults to 4. The standard deviation is multiplied by this value to determine the upper and lower limits on the chart

Value

data.frame showing original inputs and calculated control limits

Examples

```
test_vec3 <- c(1,1,2,3,5,7,11,7,5,7,8,9,5)
controls <- cusum_control(test_vec3, target = 4)
```

`cusum_control_median` *Calculate cusum control limits centred on target value or median*

Description

Calculates cusum statistics and control limits based on input parameters. If no target value is supplied the median of the x value will be used.

Usage

```
cusum_control_median(
  x,
  target = NULL,
  std_dev = NULL,
  desired_shift = 1,
  k = 0.5,
  h = 4
)
```

Arguments

`x`	input vector
`target`	target value for comparison, the median of x will be used if missing
`std_dev`	Defaults to the screened moving range of x. A known or desired value for standard deviation can be supplied instead.
`desired_shift`	how many standard deviations do you want to detect? This value is typically between 0.5 to 1. Defaults to 1.
`k`	allowable slack - defaults to half the standard deviation multiplied by desired shift
`h`	action limits - usually between 4 and 5, defaults to 4. The standard deviation is multiplied by this value to determine the upper and lower limits on the chart

Value

data.frame showing original inputs and calculated control limits

Examples

```
test_vec3 <- c(1,1,2,3,5,7,11,7,5,7,8,9,5)
controls <- cusum_control_median(test_vec3, target = 4)
controls_median <- cusum_control_median(test_vec3)
```

`cusum_control_plot` *Cumulative sum control charts as small multiples*

Description

Returns a ggplot2 object showing calculated control limits. This function should be used after calculating the control limits with either `cusum_control` or `cusum_control_median`.

Usage

```
cusum_control_plot(
  df,
  xvar,
  show_below = FALSE,
  pos_col = "#385581",
  centre_col = "black",
  neg_col = "#6dbac6",
  highlight_col = "#c9052c",
  facet_var = NULL,
  facet_scales = "free_y",
  scale_type = NULL,
  datebreaks = NULL,
  title_text = NULL,
  ...
)
```

Arguments

`df`	input data frame generated by cusum_control function
`xvar`	the variable on the x axis, typically an obervation number or date/time
`show_below`	whether to highlight points below the LCL, default is FALSE
`pos_col`	line and point colour for positive values
`centre_col`	line colour for centre line
`neg_col`	line nd point colour for negative values
`highlight_col`	• point colour for values outside UCL and (optionally) LCL
`facet_var`	• the grouping variable to facet the charts by. If not supplied a non faceted plot is generated
`facet_scales`	defaults to "free_y", but any of the usual ggplot2 facet values can be supplied e.g. "fixed" or "free_x"
`scale_type`	if you need a date or datetime scale, specify either "date" or "datetime" here. Otherwise, leave as NULL and ggplot2 will pick an appropriate scale for you
`datebreaks`	a character string specifying the breaks as text e.g "2 days" or "3 weeks". See ggplot2 date_breaks for further details
`title_text`	optional title for chart
`...`	further arguments passed on to ggplot2

Value

ggplot2 object suited for further amendments if required.

Examples

```
test_vec3 <- c(1,1,2,3,5,7,11,7,5,7,8,9,5)
controls <- cusum_control(test_vec3, target = 4)
cusum_control_plot(controls, xvar = obs)
```

---

cusum_single *Calculates the cumulative sum statistic relative to target, or mean value*

Description

Calculates the cumulative sum statistic of a vector of values, centred on either the mean of the data, or a supplied target value.

Usage

```
cusum_single(x, target = NULL)
```

Arguments

x	a numeric vector from which to calculate the cumulative sum statistics
target	value to compare each element of x to. If not provided, the mean of x will be calculated and used as a target value

Value

a vector of the cumulative sum statistic, centred on the target value

Examples

```
test_vec <- c(0.175, 0.152, 0.15, 0.207, 0.136, 0.212, 0.166)
cusum_single(test_vec)
```

cusum_single_df *Cumulative sum statistics and variances, centred on target value or mean*

Description

Provides substantially more information than cusum_single. Outputs a data.frame with the original values, target, the variance, the cumulative sum of the variance, and the cumulative sum centered on the target value. This centering is achieved by adding the target value to the cumulative sum.

Usage

```
cusum_single_df(x, target = NULL)
```

Arguments

x	a numeric vector from which to calculate the cumulative sum statistics
target	value to compare each element of x to. If not provided, the mean of x will be calculated and used as a target value

Value

a dataframe with the original values, target, the variance, the cumulative sum of the variance, and the cumulative sum centered on the target value. This centering is achieved by adding the target value to the cumulative sum.

Examples

```
test_vec <- c(0.175, 0.152, 0.15, 0.207, 0.136, 0.212, 0.166)
cusum_single_df(test_vec, target = 0.16)
```

cusum_single_median *Calculates the cumulative sum statistic relative to target, or median value*

Description

Calculates the cumulative sum statistic of a vector of values, centred on either the median of the data, or a supplied target value.

Usage

```
cusum_single_median(x, target = NULL)
```

Arguments

x	a numeric vector from which to calculate the cumulative sum statistics
target	value to compare each element of x to. If not provided, the median value of x will be calculated and used as a target value

Value

a vector of the cumulative sum statistic, centred on the target value

Examples

```
test_vec <- c(0.175, 0.152, 0.15, 0.207, 0.136, 0.212, 0.166)
cusum_single_median(test_vec)
```

---

cusum_single_median_df

Cumulative sum statistics and variances, centred on target value or median

---

Description

Provides substantially more information than cusum_single_median. Outputs a data.frame with the original values, target, the variance, the cumulative sum of the variance, and the cumulative sum centered on the target value. This centering is achieved by adding the target value to the cumulative sum.

Usage

```
cusum_single_median_df(x, target = NULL)
```

Arguments

x	a numeric vector from which to calculate the cumulative sum statistics
target	value to compare each element of x to. If not provided, the median value of x will be calculated and used as a target value

Value

a dataframe with the original values, target, the variance, the cumulative sum of the variance, and the cumulative sum centered on the target value. This centering is achieved by adding the target value to the cumulative sum.

Examples

```
test_vec <- c(0.175, 0.152, 0.15, 0.207, 0.136, 0.212, 0.166)
cusum_single_median_df(test_vec, target = 0.16)
cusum_single_median_df(test_vec)
```

Index