

Package ‘cusumcharter’

July 22, 2025

Title Easier CUSUM Control Charts

Version 0.1.0

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

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

NeedsCompilation no

Author John MacKintosh [aut, cre]

Maintainer John MacKintosh <johnmackintosh.jm@gmail.com>

Repository CRAN

Date/Publication 2021-11-15 08:50:02 UTC

Contents

cusum_control	2
cusum_control_median	3
cusum_control_plot	4
cusum_single	5
cusum_single_df	6
cusum_single_median	6
cusum_single_median_df	7

Index	8
--------------	----------

cusum_control	<i>cusum_control</i>
---------------	----------------------

Description

cusum_control

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` *cusum_control_median*

Description

`cusum_control_median`

Usage

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

Arguments

<code>x</code>	input vector
<code>target</code>	target value for comparison, the median of <code>x</code> will be used if missing
<code>std_dev</code>	Defaults to the screened moving range of <code>x</code> . A known or desired value for standard deviation can be supplied instead.
<code>desired_shift</code>	how many standard deviations do you want to detect? This value is typically between 0.5 to 1. Defaults to 1.
<code>k</code>	allowable slack - defaults to half the standard deviation multiplied by desired shift
<code>h</code>	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	<i>cusum_control_plot</i>
--------------------	---------------------------

Description

cusum_control_plot

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 observation 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 and 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`

cusum_single

Description

`cusum_single`

Usage

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

Arguments

<code>x</code>	a numeric vector from which to calculate the cumulative sum statistics
<code>target</code>	value to compare each element of <code>x</code> to. If not provided, the mean of <code>x</code> 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	<i>cusum_single_df</i>
-----------------	------------------------

Description

cusum_single_df

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	<i>cusum_single_median</i>
---------------------	----------------------------

Description

cusum_single_median

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`

cusum_single_median_df

Description

`cusum_single_median_df`

Usage

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

Arguments

<code>x</code>	a numeric vector from which to calculate the cumulative sum statistics
<code>target</code>	value to compare each element of <code>x</code> to. If not provided, the median value of <code>x</code> 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

cusum_control, [2](#)
cusum_control_median, [3](#)
cusum_control_plot, [4](#)
cusum_single, [5](#)
cusum_single_df, [6](#)
cusum_single_median, [6](#)
cusum_single_median_df, [7](#)