

Package ‘ggQC’

March 29, 2017

Type Package

Title Quality Control Charts for the Grammar of Graphics Plotting System

Version 0.0.1

Author Kenith Grey

Maintainer Kenith Grey <kenithgrey@r-bar.net>

Description Plot single and faceted type quality control charts within the grammar of graphics plotting framework.

Depends R (>= 2.10)

License GPL-3 | file LICENSE

Encoding UTF-8

LazyData true

Suggests testthat, gridExtra, knitr, rmarkdown

RoxygenNote 6.0.0

Imports ggplot2, reshape2, stats, plyr

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2017-03-29 06:41:34 UTC

R topics documented:

cBar_LCL	2
cBar_UCL	3
mR	4
mR_points	4
mR_UCL	5
npBar	6
npBar_LCL	6
npBar_UCL	7

pBar	8
pBar_LCL	9
pBar_UCL	9
QCrang	10
QC_Lines	11
rBar	14
rBar_LCL	14
rBar_UCL	15
rMedian	16
rMedian_LCL	17
rMedian_UCL	17
sBar	18
sBar_LCL	19
sBar_UCL	19
stat_mR	20
stat_QC	22
stat_QC_labels	25
uBar	28
uBar_LCL	29
uBar_UCL	29
xBar_Bar	30
xBar_one_LCL	31
xBar_one_UCL	31
xBar_rBar_LCL	32
xBar_rBar_UCL	33
xBar_rMedian_LCL	34
xBar_rMedian_UCL	35
xBar_sBar_LCL	36
xBar_sBar_UCL	37
xMedian_Bar	38
xMedian_rBar_LCL	38
xMedian_rBar_UCL	39
xMedian_rMedian_LCL	40
xMedian_rMedian_UCL	41

Index	42
--------------	-----------

cBar_LCL	<i>Lower Control Limit: Count Data (c-chart)</i>
----------	--

Description

Calculates lower control limit (LCL) for count data acquired over the same-sized area of opportunity. Negative values are reported as 0.

Usage

```
cBar_LCL(y, na.rm = FALSE, ...)
```

Arguments

<code>y</code>	Vector of count data. Each observation having the same-area of opportunity.
<code>na.rm</code>	a logical value indicating whether NA values should be stripped before the computation proceeds.
<code>...</code>	further arguments passed to or from other methods.

Value

A number; 3-sigma lower control limit (LCL). Function returns 0 for negative values.

Examples

```
set.seed(5555)
y <- rpois(30, 9)
cBar_LCL(y)
```

cBar_UCL

Upper Control Limit: Count Data (c-chart)

Description

Calculates upper control limit (UCL) for count data acquired over the same-sized area of opportunity.

Usage

```
cBar_UCL(y, na.rm = FALSE, ...)
```

Arguments

<code>y</code>	Vector of count data. Each observation having the same-area of opportunity.
<code>na.rm</code>	a logical value indicating whether NA values should be stripped before the computation proceeds.
<code>...</code>	further arguments passed to or from other methods.

Value

A number; 3-sigma upper control limit (UCL)

Examples

```
set.seed(5555)
y <- rpois(30, 9)
cBar_UCL(y)
```

mR	<i>Mean One-Point Moving Range</i>
----	------------------------------------

Description

Calculates the mean one-point moving range used when constructing a moving-range chart.

Usage

```
mR(y, na.rm = TRUE, ...)
```

Arguments

y	Vector of values
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to or from other methods.

Value

A number; mean one-point moving range.

Examples

```
set.seed(5555)
values <- rnorm(n = 100, mean = 25, sd = 1)
mR(values)
```

mR_points	<i>One Point Moving Range of Vector</i>
-----------	---

Description

Calculates a one-point moving range vector given an input vector of values. Output often used to produce mR-chart.

Usage

```
mR_points(y)
```

Arguments

y	: vector of values
---	--------------------

Value

Vector of one-point moving range.

Examples

```
y <- seq(-5:5)
mR_points(y)
```

mR_UCL

Mean One-Point Moving Range Upper Control Limit (UCL)

Description

Calculates the mean one-point moving range UCL used when constructing a moving-range chart.

Usage

```
mR_UCL(y, na.rm = FALSE, ...)
```

Arguments

y	Vector of values
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to or from other methods.

Value

A number; mean one-point moving range UCL.

Examples

```
set.seed(5555)
values <- rnorm(n = 100, mean = 25, sd = 1)
mR_UCL(values)
```

npBar	<i>Mean Value: Binomial Data (np-chart)</i>
-------	---

Description

Calculates the mean value for binomial count data acquired over the same-sized area of opportunity.

Usage

```
npBar(y, n, na.rm = FALSE, ...)
```

Arguments

y	Vector of binomial count data (not proportions). Each observation having the same-area of opportunity.
n	A number representing the area of opportunity.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to or from other methods.

Value

A number; mean value

Examples

```
set.seed(5555)
p <- rbinom(n = 100, size = 30, prob = .2)
npBar(y = p, n = 30)
```

npBar_LCL	<i>Lower Control Limit: Binomial Data (np-chart)</i>
-----------	--

Description

Calculates lower control limit (LCL) for binomial count data acquired over the same-sized area of opportunity.

Usage

```
npBar_LCL(y, n, na.rm = FALSE, ...)
```

Arguments

y	Vector of binomial count data (not proportions). Each observation having the same-area of opportunity.
n	A number representing the area of opportunity.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to or from other methods.

Value

A number; 3-sigma upper control limit (LCL)

Examples

```
set.seed(5555)
p <- rbinom(n = 100, size = 30, prob = .2)
npBar_LCL(y = p, n = 30)
```

npBar_UCL

*Upper Control Limit: Binomial Data (np-chart)***Description**

Calculates upper control limit (UCL) for binomial count data acquired over the same-sized area of opportunity.

Usage

```
npBar_UCL(y, n, na.rm = FALSE, ...)
```

Arguments

y	Vector of binomial count data (not proportions). Each observation having the same-area of opportunity.
n	A number representing the area of opportunity.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to or from other methods.

Value

A number; 3-sigma upper control limit (UCL)

Examples

```
set.seed(5555)
p <- rbinom(n = 100, size = 30, prob = .2)
npBar_UCL(y = p, n = 30)
```

pBar

Mean Proportion: Binomial Data (p-chart)

Description

Calculates overall mean proportion for binomial proportion data acquired over a variable area of opportunity.

Usage

```
pBar(y, n, na.rm = FALSE, ...)
```

Arguments

y	Vector of binomial proportion data (not counts). Observations may have a different area of opportunity, n.
n	A vector representing the area of opportunity.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to or from other methods.

Value

A vector of mean proportion, length equal to length of parameter y.

Examples

```
set.seed(5555)
p <- rbinom(n = 100, size = 30, prob = .2)
n <- rpois(100, 100)
pBar(y = p/n, n = n)
```


pBar_LCL

*Lower Control Limit: Binomial Data (p-chart)***Description**

Calculates point-wise lower control limit (LCL) for binomial proportion data acquired over a variable area of opportunity.

Usage

```
pBar_LCL(y, n, na.rm = FALSE, ...)
```

Arguments

y	Vector of binomial proportion data (not counts). Observations may have a different area of opportunity, n.
n	A vector representing the area of opportunity.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to or from other methods.

Value

A vector; point-wise 3-sigma lower control limit (LCL)

Examples

```
set.seed(5555)
p <- rbinom(n = 100, size = 30, prob = .2)
n <- rpois(100, 100)
pBar_LCL(y = p/n, n = n)
```

pBar_UCL

*Upper Control Limit: Binomial Data (p-chart)***Description**

Calculates point-wise upper control limit (UCL) for binomial proportion data acquired over a variable area of opportunity.

Usage

```
pBar_UCL(y, n, na.rm = FALSE, ...)
```

Arguments

<code>y</code>	Vector of binomial proportion data (not counts). Observations may have a different area of opportunity, <code>n</code> .
<code>n</code>	A vector representing the area of opportunity.
<code>na.rm</code>	a logical value indicating whether NA values should be stripped before the computation proceeds.
<code>...</code>	further arguments passed to or from other methods.

Value

A vector; point-wise 3-sigma upper control limit (UCL)

Examples

```
set.seed(5555)
p <- rbinom(n = 100, size = 30, prob = .2)
n <- rpois(100, 100)
pBar_UCL(y = p/n, n = n)
```

QCrange

Range: Max Min Difference

Description

Given a set of numbers, function calculates the difference between the maximum and minimum value.

Usage

```
QCrange(y)
```

Arguments

`y` : vector of values

Value

a number.

Examples

```
y <- seq(-5:5)
QCrange(y)
```

QC_Lines

*Calculate QC Limits***Description**

Calculates QC chart lines for the following chart types and reports in a dataframe:

- **Individuals Charts:** mR, XmR,
- **Attribute Charts:** c, np, p, u,
- **Studentized Charts:** xBar.rBar, xBar.rMedian, xBar.sBar, xMedian.rBar, xMedian.rMedian,
- **Dispersion Charts:** rBar, rMedian, sBar.

Usage

```
QC_Lines(data = NULL, value = NULL, grouping = NULL, formula = NULL,
         n = NULL, method = "xBar.rBar", na.rm = FALSE)
```

Arguments

data	vector or dataframe, as indicated below for each chart type <ul style="list-style-type: none"> • Individuals & Attribute Charts: vector of values; • Studentized & Dispersion Charts: dataframe
value	Studentized Charts and Dispersion Charts , numeric vector in dataframe with values of interest
grouping	Studentized Charts and Dispersion Charts : single factor/variable to split the dataframe "values" by
formula	Studentized Charts and Dispersion Charts : a formula, such as $y \sim x1 + x2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables
n	number or vector as indicated below for each chart type. <ul style="list-style-type: none"> • Individuals Charts: No effect • Attribute Charts: (p and u) vector, indicating sample area of opportunity. • Attribute Charts: (np) number, indicating constant sampling area of opportunity. • Studentized Charts: number, user specified subgroup size. • Dispersion Charts: No effect
method	string, calling the following methods: <ul style="list-style-type: none"> • Individuals Charts: mR, XmR, • Attribute Charts: c, np, p, u, • Studentized Charts: xBar.rBar, xBar.rMedian, xBar.sBar, xMedian.rBar, xMedian.rMedian • Dispersion Charts: rBar, rMedian, sBar.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.

Value

a dataframe,

- **Attribute Data:** (p and u) Center Line, Upper Control Limit and Lower Control limit for each point.
- **Other Data:** single line dataframe, with relevant control limits noted in column headings.

Note

If using the **formula** argument do not use **value** and **group** arguments.

References

Wheeler, DJ, and DS Chambers. Understanding Statistical Process Control, 2nd Ed. Knoxville, TN: SPC, 1992. Print.

Examples

```
#####
# Example 1: Charts other than "p" or "u" #
#####

# Load Libraries -----
require(ggQC)
require(plyr)
require(ggplot2)

# Setup Data -----
set.seed(5555)
Process1 <- data.frame(processID = as.factor(rep(1,100)),
                      metric_value = rnorm(100,0,1),
                      subgroup_sample=rep(1:20, each=5),
                      Process_run_id = 1:100)

set.seed(5555)
Process2 <- data.frame(processID = as.factor(rep(2,100)),
                      metric_value = rnorm(100,5, 1),
                      subgroup_sample=rep(1:10, each=10),
                      Process_run_id = 101:200)

Both_Processes <- rbind(Process1, Process2)

# QC Values For Individuals -----
# All Together
QC_Lines(data = Both_Processes$metric_value, method = "XmR")

# For Each Process
ddply(Both_Processes, .variables = "processID",
      .fun =function(df){
        QC_Lines(data = df$metric_value, method = "XmR")
      }
    )
```

```

    )

# QC Values For Studentized Runs-----
# All Together
  QC_Lines(data = Both_Processes,
    formula = metric_value ~ subgroup_sample)

# For Each Process
  ddply(Both_Processes, .variables = "processID",
    .fun =function(df){
      QC_Lines(data = df, formula = metric_value ~ subgroup_sample)
    }
  )

#####
# Example 2 "p" data #
#####

# Setup p Data -----
set.seed(5555)
bin_data <- data.frame(
  trial = 1:30,
  Num_Incomplete_Items = rpois(n = 30, lambda = 30),
  Num_Items_in_Set = runif(n = 30, min = 50, max = 100))

bin_data$Proportion_Incomplete <- bin_data$Num_Incomplete_Items/bin_data$Num_Items_in_Set

# QC_Lines for "p" data -----
  QC_Lines(data = bin_data$Proportion_Incomplete,
    n = bin_data$Num_Items_in_Set, method="p")

#####
# Example 3 "u" data #
#####

# Setup u Data -----
set.seed(5555)
bin_data <- data.frame(
  trial=1:30,
  Num_of_Blemishes = rpois(n = 30, lambda = 30),
  Num_Items_Inspected = runif(n = 30, min = 50, max = 100))

bin_data$Blemish_Rate <- bin_data$Num_of_Blemishes/bin_data$Num_Items_Inspected

# QC Lines for "u" data -----
  QC_Lines(data = bin_data$Blemish_Rate,
    n = bin_data$Num_Items_Inspected, method="u")

```

rBar	Mean Subgroup Range
------	---------------------

Description

Calculates the mean subgroup range used when constructing a XbarR chart.

Usage

```
rBar(data, value, grouping, formula = NULL, ...)
```

Arguments

- data data frame to be processed
- value numeric vector in a data frame with values of interest.
- grouping single factor/variable to split the data frame "values" by.
- formula a formula, such as y ~ x1 + x2, where the y variable is numeric data to be split into groups according to the grouping x factors/variables
- ... further arguments passed to or from other methods.

Value

A number; mean subgroup range.

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
rBar(data = df, formula = v~g)
```

rBar_LCL	Mean Subgroup Range Lower Control Limit (LCL)
----------	---

Description

Calculates the mean subgroup range Lower control limit (UCL) used when constructing a XbarR chart.

Usage

```
rBar_LCL(data = data, value = value, grouping = grouping,
          formula = NULL, ...)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
formula	a formula, such as $y \sim x1 + x2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables
...	further arguments passed to or from other methods.

Value

A number; mean subgroup range lower control limit (LCL).

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
rBar_LCL(data = df, formula = v~g)
```

rBar_UCL

*Mean Subgroup Range Upper Control Limit (UCL)***Description**

Calculates the mean subgroup range upper control limit (UCL) used when constructing a XbarR chart.

Usage

```
rBar_UCL(data = data, value = value, grouping = grouping,
  formula = NULL, ...)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
formula	a formula, such as $y \sim x1 + x2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables
...	further arguments passed to or from other methods.

Value

A number; mean subgroup range upper control limit (UCL).

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
rBar_UCL(data = df, formula = v~g)
```

rMedian	<i>Median of Subgroup Ranges</i>
---------	----------------------------------

Description

Calculates the median of subgroup ranges, used when constructing xBar_rMedian charts.

Usage

```
rMedian(data, value, grouping, formula = NULL, ...)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
formula	a formula, such as y ~ x1 + x2, where the y variable is numeric data to be split into groups according to the grouping x factors/variables
...	further arguments passed to or from other methods.

Value

A number; median subgroup range.

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
rMedian(data = df, formula = v~g)
```

rMedian_LCL	<i>Median of Subgroup Ranges Lower Control Limit (LCL)</i>
-------------	--

Description

Calculates the median of subgroup range Lower control limit (LCL) used when constructing a xBar_rMedian chart.

Usage

```
rMedian_LCL(data = data, value = value, grouping = grouping,
             formula = NULL, ...)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
formula	a formula, such as $y \sim x1 + x2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables
...	further arguments passed to or from other methods.

Value

A number; median of subgroup range lower control limit (LCL).

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
rMedian_LCL(data = df, formula = v~g)
```

rMedian_UCL	<i>Median of Subgroup Ranges Upper Control Limit (UCL)</i>
-------------	--

Description

Calculates the median of subgroup range upper control limit (UCL) used when constructing a xBar_rMedian chart.

Usage

```
rMedian_UCL(data = data, value = value, grouping = grouping,
             formula = NULL, ...)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
formula	a formula, such as $y \sim x_1 + x_2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables
...	further arguments passed to or from other methods.

Value

A number; median of subgroup range upper control limit (UCL).

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
rMedian_UCL(data = df, formula = v~g)
```

sBar

Mean Subgroup Standard Deviation

Description

Calculates the mean subgroup standard deviation used when constructing a XbarS chart.

Usage

```
sBar(data, value, grouping, formula = NULL, ...)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
formula	a formula, such as $y \sim x_1 + x_2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables
...	further arguments passed to or from other methods.

Value

A number; mean subgroup standard deviation.

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
sBar(data = df, formula = v~g)
```

sBar_LCL*Mean Subgroup Standard Deviation Lower Control Limit (LCL)*

Description

Calculates the mean subgroup standard deviation Lower control limit (UCL) used when constructing a XbarS chart.

Usage

```
sBar_LCL(data = data, value = value, grouping = grouping,  
          formula = NULL, ...)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
formula	a formula, such as $y \sim x1 + x2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables
...	further arguments passed to or from other methods.

Value

A number; mean subgroup standard deviation lower control limit (LCL).

Examples

```
set.seed(5555)  
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))  
sBar_LCL(data = df, formula = v~g)
```

sBar_UCL*Mean Subgroup Standard Deviation Upper Control Limit (UCL)*

Description

Calculates the mean subgroup standard deviation upper control limit (UCL) used when constructing a XbarS chart.

Usage

```
sBar_UCL(data = data, value = value, grouping = grouping,  
          formula = NULL, ...)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
formula	a formula, such as $y \sim x_1 + x_2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables
...	further arguments passed to or from other methods.

Value

A number; mean subgroup standard deviation upper control limit (UCL).

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
sBar_UCL(data = df, formula = v~g)
```

stat_mR

*Generate mR chart in ggplot***Description**

ggplot stat used to create a mR chart in ggplot

Usage

```
stat_mR(mapping = NULL, data = NULL, geom = "point",
  position = "identity", show.legend = NA, inherit.aes = TRUE,
  na.rm = FALSE, color.mr_point = "black", color.mr_line = "black",
  color.qc_limits = "red", color.qc_center = "blue", ...)
```

Arguments

mapping	Set of aesthetic mappings created by aes or aes_ . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	<p>The data to be displayed in this layer. There are three options:</p> <p>If <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to ggplot.</p> <p>A <code>data.frame</code>, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify for which variables will be created.</p> <p>A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data.</p>

geom	The geometric object to use display the data
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders .
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
color.mr_point	color, to be used for the mR points.
color.mr_line	color, to be used for line connecting points.
color.qc_limits	color, used to colorize the plot's upper and lower mR control limits.
color.qc_center	color, used to colorize the plot's center line.
...	other arguments passed on to layer . These are often aesthetics, used to set an aesthetic to a fixed value, like color = "red" or size = 3. They may also be parameters to the paired geom/stat.

Value

data need to produce the mR plot in ggplot.

Examples

```
#####
# Example 1: mR Chart #
#####

# Load Libraries -----
require(ggQC)
require(ggplot2)

# Setup Data -----
set.seed(5555)
Process1 <- data.frame(processID = as.factor(rep(1,100)),
                        metric_value = rnorm(100,0,1),
                        subgroup_sample=rep(1:20, each=5),
                        Process_run_id = 1:100)

set.seed(5556)
Process2 <- data.frame(processID = as.factor(rep(2,100)),
                        metric_value = rnorm(100,5, 1),
                        subgroup_sample=rep(1:10, each=10),
                        Process_run_id = 101:200)

Both_Processes <- rbind(Process1, Process2)
```

```
# One Plot Both Processes -----
ggplot(Both_Processes, aes(x=Process_run_id, y = metric_value)) +
  stat_mR() + ylab("Moving Range")

# Facet Plot - Both Processes -----
ggplot(Both_Processes, aes(x=Process_run_id, y = metric_value)) +
  stat_mR() + ylab("Moving Range") +
  facet_grid(~processID, scales = "free_x")
```

stat_QC

Produce QC Charts with ggplot Framework.

Description

Produce QC charts with ggplot framework. Support for faceting and layering of multiple QC chart lines on a single plot. Charts supported (see method argument for call):

- **Individuals Charts:** mR, XmR,
- **Attribute Charts:** c, np, p, u,
- **Studentized Charts:** xBar.rBar, xBar.rMedian, xBar.sBar, xMedian.rBar, xMedian.rMedian,
- **Dispersion Charts:** rBar, rMedian, sBar.

To label chart lines see [stat_QC_labels](#)

Usage

```
stat_QC(mapping = NULL, data = NULL, geom = "hline",
  position = "identity", na.rm = FALSE, show.legend = NA,
  inherit.aes = TRUE, n = NULL, method = "xBar.rBar",
  color.qc_limits = "red", color.qc_center = "green", ...)
```

Arguments

mapping	Set of aesthetic mappings created by aes or aes_ . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	<p>The data to be displayed in this layer. There are three options:</p> <p>If <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to ggplot.</p> <p>A <code>data.frame</code>, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify for which variables will be created.</p> <p>A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data.</p>
geom	The geometric object to use display the data

position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders .
n	number, for <ul style="list-style-type: none"> • Studentized Charts, used for custom or hypothetical subgroup size. • np Charts, used to specify a fixed area of opportunity.
method	string, calling the following methods: <ul style="list-style-type: none"> • Individuals Charts: mR, XmR, • Attribute Charts: c, np, p, u, • Studentized Charts: xBar.rBar, xBar.rMedian, xBar.sBar, xMedian.rBar, xMedian.rMedian • Dispersion Charts: rBar, rMedian, sBar.
color.qc_limits	color, used to colorize the plot's upper and lower mR control limits.
color.qc_center	color, used to colorize the plot's center line.
...	other arguments passed on to layer . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>color = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.

Value

data need to produce the mR plot in ggplot.

Examples

```
# Load Libraries -----
require(ggQC)
require(ggplot2)

# Setup Data -----
set.seed(5555)
Process1 <- data.frame(processID = as.factor(rep(1,100)),
                      metric_value = rnorm(100,0,1),
                      subgroup_sample = rep(1:20, each=5),
                      Process_run_id = 1:100)

set.seed(5556)
Process2 <- data.frame(processID = as.factor(rep(2,100)),
                      metric_value = rnorm(100,5, 1),
                      subgroup_sample = rep(1:10, each=10),
                      Process_run_id = 101:200)
```

```

Both_Processes <- rbind(Process1, Process2)

#####
# Example 1: XmR Chart      #
#####

ggplot(Both_Processes, aes(x=Process_run_id, y = metric_value)) +
  geom_point() + geom_line() + stat_QC(method="XmR") +
  stat_QC_labels(method="XmR", digits = 2) +
  facet_grid(.~processID, scales = "free_x")

ggplot(Both_Processes, aes(x=Process_run_id, y = metric_value)) +
  stat_mR() + ylab("Moving Range") +
  stat_QC_labels(method="mR", digits = 2) +
  facet_grid(.~processID, scales = "free_x")

#####
# Example 2: XbarR Chart   #
#####

ggplot(Both_Processes, aes(x = subgroup_sample,
                           y = metric_value,
                           group = processID)) +
  stat_summary(fun.y = "mean", color = "blue", geom = c("point")) +
  stat_summary(fun.y = "mean", color = "blue", geom = c("line")) +
  stat_QC(method = "xBar.rBar") + facet_grid(.~processID, scales = "free_x")

ggplot(Both_Processes, aes(x = subgroup_sample,
                           y = metric_value,
                           group = processID)) +
  stat_summary(fun.y = "QCrange", color = "blue", geom = "point") +
  stat_summary(fun.y = "QCrange", color = "blue", geom = "line") +
  stat_QC(method = "rBar") +
  ylab("Range") +
  facet_grid(.~processID, scales = "free_x")

#####
# Example 3: p Chart      #
#####
# p chart Setup -----
set.seed(5556)
bin_data <- data.frame(
  trial=1:30,
  Num_Incomplete_Items = rpois(30, lambda = 30),
  Num_Items_in_Set = runif(n = 30, min = 50, max = 100))
bin_data$Proportion_Incomplete <- bin_data$Num_Incomplete_Items/bin_data$Num_Items_in_Set

# Plot p chart -----
ggplot(data = bin_data, aes(x=trial,

```



```

                                y=Proportion_Incomplete,
                                n=Num_Items_in_Set)) +
geom_point() + geom_line() +
stat_QC(method = "p")

#####
# Example 4: u Chart      #
#####
# u chart Setup -----
set.seed(5555)
bin_data <- data.frame(
  trial=1:30,
  Num_of_Blemishes = rpois(30, lambda = 30),
  Num_Items_Inspected = runif(n = 30, min = 50, max = 100)
)
bin_data$Blemish_Rate <- bin_data$Num_of_Blemishes/bin_data$Num_Items_Inspected

# Plot u chart -----
ggplot(data = bin_data, aes(x=trial,
                             y=Blemish_Rate,
                             n=Num_Items_Inspected)) +
geom_point() + geom_line() +
stat_QC(method = "u")

#####
# Example 5: np Chart    #
#####
# np chart Setup -----
set.seed(5555)
bin_data <- data.frame(
  trial=1:30,
  NumNonConforming = rbinom(30, 30, prob = .50))
Units_Tested_Per_Batch <- 60

# Plot np chart -----
ggplot(data = bin_data, aes(trial, NumNonConforming)) +
geom_point() +
stat_QC(method = "np", n = Units_Tested_Per_Batch)

```

stat_QC_labels

Write QC Line Lables to ggplot QC Charts.

Description

Write QC line lables to ggplot QC Charts. Useful if you want to see the value of the center line and QC limits. see method argument for methods supported.

Usage

```
stat_QC_labels(mapping = NULL, data = NULL, geom = "label",
  position = "identity", na.rm = FALSE, show.legend = NA,
  inherit.aes = TRUE, n = NULL, digits = 1, method = "xBar.rBar",
  color.qc_limits = "red", color.qc_center = "black", text.size = 3, ...)
```

Arguments

mapping	Set of aesthetic mappings created by aes or aes_ . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	<p>The data to be displayed in this layer. There are three options:</p> <p>If <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to ggplot.</p> <p>A <code>data.frame</code>, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify for which variables will be created.</p> <p>A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data.</p>
geom	The geometric object to use display the data
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
show.legend	logical. Should this layer be included in the legends? <code>NA</code> , the default, includes if any aesthetics are mapped. <code>FALSE</code> never includes, and <code>TRUE</code> always includes.
inherit.aes	If <code>FALSE</code> , overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders .
n	<p>number, for</p> <ul style="list-style-type: none"> • Studentized Charts, used for custom or hypothetical subgroup size. • np Charts, used to specify a fixed area of opportunity.
digits	integer, indicating the number of decimal places
method	<p>string, calling the following methods:</p> <ul style="list-style-type: none"> • Individuals Charts: <code>mR</code>, <code>XmR</code>, • Attribute Charts: <code>c</code>, <code>np</code>, <code>p</code>, <code>u</code>, • Studentized Charts: <code>xBar.rBar</code>, <code>xBar.rMedian</code>, <code>xBar.sBar</code>, <code>xMedian.rBar</code>, <code>xMedian.rMedian</code> • Dispersion Charts: <code>rBar</code>, <code>rMedian</code>, <code>sBar</code>.
color.qc_limits	color, used to colorize the plot's upper and lower mR control limits.
color.qc_center	color, used to colorize the plot's center line.

text.size number, size of the text label

... other arguments passed on to [layer](#). These are often aesthetics, used to set an aesthetic to a fixed value, like color = "red" or size = 3. They may also be parameters to the paired geom/stat.

Value

data need to produce the mR plot in ggplot.

Examples

```
#####
# Example 1: mR Chart #
#####

# Load Libraries -----
require(ggQC)
require(ggplot2)

# Setup Data -----
set.seed(5555)
Process1 <- data.frame(processID = as.factor(rep(1,100)),
                        metric_value = rnorm(100,0,1),
                        subgroup_sample=rep(1:20, each=5),
                        Process_run_id = 1:100)

set.seed(5556)
Process2 <- data.frame(processID = as.factor(rep(2,100)),
                        metric_value = rnorm(100,5, 1),
                        subgroup_sample=rep(1:10, each=10),
                        Process_run_id = 101:200)

Both_Processes <- rbind(Process1, Process2)

# Facet Plot - Both Processes -----
ggplot(Both_Processes, aes(x=Process_run_id, y = metric_value)) +
  geom_point() + geom_line() + stat_QC(method="XmR") +
  stat_QC_labels(method="XmR", digits = 2) +
  facet_grid(.~processID, scales = "free_x")

ggplot(Both_Processes, aes(x=Process_run_id, y = metric_value)) +
  stat_mR() + ylab("Moving Range") +
  stat_QC_labels(method="mR", digits = 2) +
  facet_grid(.~processID, scales = "free_x")

#####
# Example 2: XbarR Chart #
#####
# Facet Plot - Studentized Process -----

ggplot(Both_Processes, aes(x=subgroup_sample,
                           y = metric_value,
```

```
      group = processID)) +
  geom_point(alpha=.2) +
  stat_summary(fun.y = "mean", color="blue", geom=c("point")) +
  stat_summary(fun.y = "mean", color="blue", geom=c("line")) +
  stat_QC() + facet_grid(.~processID, scales = "free_x") +
  stat_QC_labels(text.size =3, label.size=.1)

ggplot(Both_Processes, aes(x=subgroup_sample,
      y = metric_value,
      group = processID)) +
  stat_summary(fun.y = "QCrange", color="blue", geom = "point") +
  stat_summary(fun.y = "QCrange", color="blue", geom = "line") +
  stat_QC(method="rBar") +
  stat_QC_labels(digits=2, method="rBar") +
  ylab("Range") +
  facet_grid(.~processID, scales = "free_x")
```

uBar	<i>Mean Rate: Count Data (u-chart)</i>
------	--

Description

Calculates overall mean rate for count data acquired over a variable area of opportunity.

Usage

```
uBar(y, n, na.rm = FALSE, ...)
```

Arguments

- y Vector of counts per unit opportunity (rate). Observations may have a different area of opportunity, n.
- n A vector representing the area of opportunity.
- na.rm a logical value indicating whether NA values should be stripped before the computation proceeds.
- ... further arguments passed to or from other methods.

Value

A vector of mean rate, length equal to length of parameter y.

Examples

```
set.seed(5555)
counts <- rpois(100, 25)
n <- rpois(100, 15)
uBar(y = counts / n, n = n)
```

uBar_LCL	<i>Lower Control Limit: Count Data (u-chart)</i>
----------	--

Description

Calculates point-wise lower control limit (LCL) for count data acquired over a variable area of opportunity.

Usage

```
uBar_LCL(y, n, na.rm = FALSE, ...)
```

Arguments

y	Vector of counts per unit opportunity (rate). Observations may have a different area of opportunity, n.
n	A vector representing the area of opportunity.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to or from other methods.

Value

A vector; point-wise 3-sigma lower control limit (LCL)

Examples

```
set.seed(5555)
counts <- rpois(100, 25)
n <- rpois(100, 15)
uBar_LCL(y = counts / n, n = n)
```

uBar_UCL	<i>Upper Control Limit: Count Data (u-chart)</i>
----------	--

Description

Calculates point-wise upper control limit (UCL) for count data acquired over a variable area of opportunity.

Usage

```
uBar_UCL(y, n, na.rm = FALSE, ...)
```

Arguments

y	Vector of counts per unit opportunity (rate). Observations may have a different area of opportunity, n.
n	A vector representing the area of opportunity.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to or from other methods.

Value

A vector; point-wise 3-sigma upper control limit (UCL)

Examples

```
set.seed(5555)
counts <- rpois(100, 25)
n <- rpois(100, 15)
uBar_UCL(y = counts / n, n = n)
```

xBar_Bar

Mean of Subgroup Means

Description

Calculates the mean subgroup means used when constructing a xBar-R or xBar-S charts.

Usage

```
xBar_Bar(data, value, grouping, formula = NULL, ...)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
formula	a formula, such as $y \sim x_1 + x_2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables
...	further arguments passed to or from other methods.

Value

A number; mean of subgroup means.

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
xBar_Bar(data = df, formula = v~g)
```

xBar_one_LCL	<i>xBar_One Lower Control Limit (LCL)</i>
--------------	---

Description

Calculates the xBar_One LCL used when constructing a xBar-One chart.

Usage

```
xBar_one_LCL(y, na.rm = FALSE, ...)
```

Arguments

y	Vector of values
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to or from other methods.

Value

A number; xBar_One Lower Control Limit (LCL)

Examples

```
set.seed(5555)
values <- rnorm(n = 100, mean = 25, sd = 1)
xBar_one_LCL(values)
```

xBar_one_UCL	<i>xBar_One Upper Control Limit (UCL)</i>
--------------	---

Description

Calculates the xBar_One UCL used when constructing a xBar-One chart.

Usage

```
xBar_one_UCL(y, na.rm = FALSE, ...)
```

Arguments

y	Vector of values
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to or from other methods.

Value

A number; xBar_One Upper Control Limit (UCL)

Examples

```
set.seed(5555)
values <- rnorm(n = 100, mean = 25, sd = 1)
xBar_one_UCL(values)
```

xBar_rBar_LCL	<i>Mean of Subgroup Means Lower Control Limit (LCL)</i>
---------------	---

Description

Calculates the mean of subgroup means lower control limit used when constructing a xBar-R charts.

Usage

```
xBar_rBar_LCL(data, value, grouping, n = NULL, natural = F,
  formula = NULL)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
n	a number indicating a hypothetical subgroup size other than, function determined subgroup n determined by the floor length of subgroup values.
natural	logical, if TRUE calculate limits for individuals (n=1) else calculate for n determined by the floor length of subgroup values
formula	a formula, such as $y \sim x_1 + x_2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables

Value

A number; mean of subgroup means lower control limit.

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
xBar_rBar_UCL(data = df, formula = v~g)
```

xBar_rBar_UCL

*Mean of Subgroup Means Upper Control Limit (UCL)***Description**

Calculates the mean of subgroup means upper control limit used when constructing a xBar-R charts.

Usage

```
xBar_rBar_UCL(data, value, grouping, n = NULL, natural = F,
  formula = NULL)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
n	a number indicating a hypothetical subgroup size other than, function determined subgroup n determined by the floor length of subgroup values.
natural	logical, if TRUE calculate limits for individuals (n=1) else calculate for n determined by the floor length of subgroup values
formula	a formula, such as $y \sim x_1 + x_2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables

Value

A number; mean of subgroup means upper control limit.

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
xBar_rBar_UCL(data = df, formula = v~g)
```

xBar_rMedian_LCL	<i>Mean of Subgroup Means Lower Control Limit (LCL) based on Median Range</i>
------------------	---

Description

Calculates the mean of subgroup means lower control limit based on the median range. The result is used when constructing a xBar-rMedian charts.

Usage

```
xBar_rMedian_LCL(data, value, grouping, n = NULL, natural = F,
  formula = NULL)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
n	a number indicating a hypothetical subgroup size other than, function determined subgroup n determined by the floor length of subgroup values.
natural	logical, if TRUE calculate limits for individuals (n=1) else calculate for n determined by the floor length of subgroup values
formula	a formula, such as $y \sim x_1 + x_2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables

Value

A number; mean of subgroup means Lower Control Limit (LCL) based on Median Range

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
xBar_rMedian_LCL(data = df, formula = v~g)
```

xBar_rMedian_UCL	<i>Mean of Subgroup Means Upper Control Limit (UCL) based on Median Range</i>
------------------	---

Description

Calculates the mean of subgroup means upper control limit based on the median range. The result is used when constructing a xBar-rMedian charts.

Usage

```
xBar_rMedian_UCL(data, value, grouping, n = NULL, natural = F,
  formula = NULL)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
n	a number indicating a hypothetical subgroup size other than, function determined subgroup n determined by the floor length of subgroup values.
natural	logical, if TRUE calculate limits for individuals (n=1) else calculate for n determined by the floor length of subgroup values
formula	a formula, such as $y \sim x_1 + x_2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables

Value

A number; mean of subgroup means Upper Control Limit (UCL) based on Median Range

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
xBar_rMedian_UCL(data = df, formula = v~g)
```

xBar_sBar_LCL	<i>Mean of Subgroup Means Lower Control Limit (LCL) based on Standard Deviation</i>
---------------	---

Description

Calculates the mean of subgroup means lower control limit based on the standard deviation. The result is used when constructing a xBar-S charts.

Usage

```
xBar_sBar_LCL(data, value, grouping, n = NULL, natural = F,
  formula = NULL)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
n	a number indicating a hypothetical subgroup size other than, function determined subgroup n determined by the floor length of subgroup values.
natural	logical, if TRUE calculate limits for individuals (n=1) else calculate for n determined by the floor length of subgroup values
formula	a formula, such as $y \sim x_1 + x_2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables

Value

A number; mean of subgroup means Lower Control Limit (LCL) based on standard deviation

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
xBar_sBar_LCL(data = df, formula = v~g)
```

xBar_sBar_UCL	<i>Mean of Subgroup Means Upper Control Limit (UCL) based on Standard Deviation</i>
---------------	---

Description

Calculates the mean of subgroup means upper control limit based on the standard deviation. The result is used when constructing a xBar-S charts.

Usage

```
xBar_sBar_UCL(data, value, grouping, n = NULL, natural = F,
  formula = NULL)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
n	a number indicating a hypothetical subgroup size other than, function determined subgroup n determined by the floor length of subgroup values.
natural	logical, if TRUE calculate limits for individuals (n=1) else calculate for n determined by the floor length of subgroup values
formula	a formula, such as $y \sim x_1 + x_2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables

Value

A number; mean of subgroup means Upper Control Limit (UCL) based on standard deviation

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
xBar_sBar_UCL(data = df, formula = v~g)
```

xMedian_Bar	<i>Mean of Subgroup Medians</i>
-------------	---------------------------------

Description

Calculates the mean of subgroup medians used when constructing a xMedian-R charts.

Usage

```
xMedian_Bar(data, value, grouping, formula = NULL, ...)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
formula	a formula, such as $y \sim x_1 + x_2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables
...	further arguments passed to or from other methods.

Value

A number; mean of subgroup medians.

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
xMedian_Bar(data = df, formula = v~g)
```

xMedian_rBar_LCL	<i>Mean of Subgroup Medians Lower Control Limit (LCL) based on Mean Range</i>
------------------	---

Description

Calculates the mean of subgroup medians lower control limit based on the mean range. The result is used when constructing a xMedian-R charts.

Usage

```
xMedian_rBar_LCL(data, value, grouping, n = NULL, natural = F,
  formula = NULL)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
n	a number indicating a hypothetical subgroup size other than, function determined subgroup n determined by the floor length of subgroup values.
natural	logical, if TRUE calculate limits for individuals (n=1) else calculate for n determined by the floor length of subgroup values
formula	a formula, such as $y \sim x_1 + x_2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables

Value

A number; mean of subgroup medians Lower Control Limit (LCL) based on mean range

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
xBar_rMedian_LCL(data = df, formula = v~g)
```

xMedian_rBar_UCL	<i>Mean of Subgroup Medians Upper Control Limit (UCL) based on mean Range</i>
------------------	---

Description

Calculates the mean of subgroup medians upper control limit based on the mean subgroup range. The result is used when constructing a xMedian-R charts.

Usage

```
xMedian_rBar_UCL(data, value, grouping, n = NULL, natural = F,
  formula = NULL)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
n	a number indicating a hypothetical subgroup size other than, function determined subgroup n determined by the floor length of subgroup values.
natural	logical, if TRUE calculate limits for individuals (n=1) else calculate for n determined by the floor length of subgroup values
formula	a formula, such as $y \sim x_1 + x_2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables

Value

A number; mean of subgroup means Upper Control Limit (UCL) based on Median Range

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
xMedian_rBar_UCL(data = df, formula = v~g)
```

xMedian_rMedian_LCL	<i>Mean of Subgroup Medians Lower Control Limit (LCL) based on Median Range</i>
---------------------	---

Description

Calculates the mean of subgroup medians lower control limit based on the median subgroup range. The result is used when constructing a xMedian-rMedian charts.

Usage

```
xMedian_rMedian_LCL(data, value, grouping, n = NULL, natural = F,
  formula = NULL)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
n	a number indicating a hypothetical subgroup size other than, function determined subgroup n determined by the floor length of subgroup values.
natural	logical, if TRUE calculate limits for individuals (n=1) else calculate for n determined by the floor length of subgroup values
formula	a formula, such as y ~ x1 + x2, where the y variable is numeric data to be split into groups according to the grouping x factors/variables

Value

A number; mean of subgroup median Lower Control Limit (LCL) based on Median Range

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
xMedian_rMedian_LCL(data = df, formula = v~g)
```

xMedian_rMedian_UCL	<i>Mean of Subgroup Medians Upper Control Limit (UCL) based on Median Range</i>
---------------------	---

Description

Calculates the mean of subgroup medians upper control limit based on the median subgroup range. The result is used when constructing a xMedian-rMedian charts.

Usage

```
xMedian_rMedian_UCL(data, value, grouping, n = NULL, natural = F,
  formula = NULL)
```

Arguments

data	data frame to be processed
value	numeric vector in a data frame with values of interest.
grouping	single factor/variable to split the data frame "values" by.
n	a number indicating a hypothetical subgroup size other than, function determined subgroup n determined by the floor length of subgroup values.
natural	logical, if TRUE calculate limits for individuals (n=1) else calculate for n determined by the floor length of subgroup values
formula	a formula, such as $y \sim x_1 + x_2$, where the y variable is numeric data to be split into groups according to the grouping x factors/variables

Value

A number; mean of subgroup median upper Control Limit (UCL) based on Median Range

Examples

```
set.seed(5555)
df <- data.frame(v=rnorm(60, 0, 1), g=rep(c("A","B","C","D","E"), each=12))
xMedian_rMedian_UCL(data = df, formula = v~g)
```

Index

aes, [20, 22, 26](#)
aes_, [20, 22, 26](#)

borders, [21, 23, 26](#)

cBar_LCL, [2](#)
cBar_UCL, [3](#)

fortify, [20, 22, 26](#)

ggplot, [20, 22, 26](#)

layer, [21, 23, 27](#)

mR, [4](#)
mR_points, [4](#)
mR_UCL, [5](#)

npBar, [6](#)
npBar_LCL, [6](#)
npBar_UCL, [7](#)

pBar, [8](#)
pBar_LCL, [9](#)
pBar_UCL, [9](#)

QC_Lines, [11](#)
QCrange, [10](#)

rBar, [14](#)
rBar_LCL, [14](#)
rBar_UCL, [15](#)
rMedian, [16](#)
rMedian_LCL, [17](#)
rMedian_UCL, [17](#)

sBar, [18](#)
sBar_LCL, [19](#)
sBar_UCL, [19](#)
stat_mR, [20](#)
stat_QC, [22](#)
stat_QC_labels, [22, 25](#)

uBar, [28](#)
uBar_LCL, [29](#)
uBar_UCL, [29](#)

xBar_Bar, [30](#)
xBar_one_LCL, [31](#)
xBar_one_UCL, [31](#)
xBar_rBar_LCL, [32](#)
xBar_rBar_UCL, [33](#)
xBar_rMedian_LCL, [34](#)
xBar_rMedian_UCL, [35](#)
xBar_sBar_LCL, [36](#)
xBar_sBar_UCL, [37](#)
xMedian_Bar, [38](#)
xMedian_rBar_LCL, [38](#)
xMedian_rBar_UCL, [39](#)
xMedian_rMedian_LCL, [40](#)
xMedian_rMedian_UCL, [41](#)