Package: rQCC (via r-universe)

September 4, 2024

Title Robust Quality Control Chart

Version 2.22.12

Date 2022-12-22

Author Chanseok Park [aut, cre]
 (<<https://orcid.org/0000-0002-2208-3498>>), Min Wang [ctb]
 (<<https://orcid.org/0000-0002-9233-7844>>)

Maintainer Chanseok Park <statpnu@gmail.com>

Depends R (>= 3.2.3)

Description Constructs various robust quality control charts based on
 the median or Hodges-Lehmann estimator (location) and the
 median absolute deviation (MAD) or Shamos estimator (scale).
 The estimators used for the robust control charts are all
 unbiased with a sample of finite size. For more details, see
 Park, Kim and Wang (2022) <[doi:10.1080/03610918.2019.1699114](doi:10.1080/03610918.2019.1699114)>.
 In addition, using this R package, the conventional quality
 control charts such as X-bar, S, R, p, np, u, c, g, h, and t
 charts are also easily constructed. This work was supported by
 the National Research Foundation of Korea (NRF) grant funded by
 the Korea government (MSIT) (No. 2022R1A2C1091319).

License GPL-2 | GPL-3

URL <https://AppliedStat.GitHub.io/R/>

BugReports <https://GitHub.com/AppliedStat/R/issues>

Encoding UTF-8

LazyData yes

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

RemoteUrl https://github.com/appliedstat/r

RemoteRef HEAD

RemoteSha bd34801eaf132f5bf389c18d2cce1e0502963bbb

Contents

---

attributes.chart.unbalanced

Attributes control chart with balanced/unbalanced samples

Description

Constructs the attributes control charts with balanced/unbalanced samples.

Usage

```
acc (x, n, type=c("p","np","c","u","g","h","t"), parameter, pEstimator=c("Wald","Wilson"),
     gEstimator = c("ML", "MVU"), tModel=c("E","W"), location.shift = 0,
     sigmaFactor=3, nk)
```

Arguments

x	a numeric vector of the number of nonconforming units.
n	a numeric vector of sample sizes.
type	a character string specifying the type of control chart.
parameter	a known Bernoulli parameter value for the corresponding chart. If not known, it is estimated. For more details, refer to vignette("acc", package="rQCC").
pEstimator	a method for estimating the Bernoulli parameter for p and np charts.
gEstimator	a method for estimating the Bernoulli parameter for g and h charts.
tModel	Probability model for t chart. "E" for Exponential and "W" for Weibull.
location.shift	a known location shift parameter value for g and h charts.

sigmaFactor	a factor for the standard deviation (σ). For example, the American Standard uses "*3*sigma*" limits (0.27% false alarm rate), while the British Standard uses "*3.09*sigma*" limits (0.20% false alarm rate).
nk	sample size for Phase-II. If nk is missing, the average of the subsample sizes is used.

Details

acc constructs the attributes control charts for nonconforming units (p and np charts) and for non-conformities per unit (c and u charts).

Value

acc returns an object of [class](#) "acc". The function summary is used to obtain and print a summary of the results and the function plot is used to plot the control chart.

Author(s)

Chanseok Park

References

ASTM (2010). *Manual on Presentation of Data and Control Chart Analysis* (STP 15-D), 8th edition. American Society for Testing and Materials, West Conshohocken, PA.

Kaminsky, F. C., J. C. Benneyan and R. D. Davis (1992). Statistical Control Charts Based on a Geometric Distribution. *Journal of Quality Technology*, **24**, 63-69.

Montgomery, D. C. (2013). *Statistical Quality Control: An Modern Introduction*, 7th edition. John Wiley & Sons, New York, NY.

Park, C. (2013). An Improved *p* Chart Based on the Wilson Interval. *Journal of Statistics and Management Systems*, **16**(2-03), 201-221.
doi:10.1080/09720510.2013.777576

Park, C. and M. Wang (2022). A study on the g and h control charts. *Communications in Statistics - Theory and Methods*, **To Appear**.
doi:10.1080/03610926.2022.2044492

Examples

```
# ==============================
# Example 1a: p chart (balanced)
# ------------------------------
# Refer to Table 31 of ASTM (2010).
x = c(1, 3, 0, 7, 2, 0, 1, 0, 8, 5, 2, 0, 1, 0, 3)
n = 400

# The conventional p chart with the balanced samples.
# Print LCL, CL, UCL.
result = acc(x, n)
print(result)
```

```
# Summary of a control chart.
summary(result)

# Plot of a control chart.
plot(result, cex.text=0.8)
text(15, 0.5, labels="p chart (with balanced sample)" )

# The p chart based on the Wilson confidence interval.
acc(x, n, pEstimator="Wilson")


# ==============================
# Example 1b: np chart (balanced)
# ------------------------------
# The data are the same as in Example 1a.
# The conventional np chart with the balanced samples.
# Print LCL, CL, UCL.
result = acc(x, n, type="np")
print(result)
summary(result)
plot(result, cex.text=0.8)
text(15, 25, labels="np chart" )

# The np chart based on the Wilson confidence interval.
acc(x, n, type="np", pEstimator="Wilson")


# ==============================
# Example 2a: p chart (unbalanced)
# ------------------------------
# Refer to Table 32 of ASTM (2010).
x = c( 9, 7, 3, 9,13,14,14,10,12,14, 6,12, 7,11, 5,
       4, 2, 4, 7, 9, 7,12, 8, 5,18, 7, 8, 8,15, 3, 5)
n = c( 580, 550, 580, 640, 880, 880, 640, 550, 580, 880,
       800, 800, 580, 580, 550, 330, 330, 640, 580, 550,
       510, 640, 300, 330, 880, 880, 800, 580, 880, 880, 330)

# The conventional p chart with the unbalanced samples.
# Print LCL, CL, UCL.
result = acc(x, n, nk=880)
print(result)

# Summary of a control chart.
summary(result)

# Plot of a control chart.
plot(result, cex.text=0.8)
text(15, 0.2, labels="p chart (with unbalanced sample)" )


# ==============================
# Example 2b: p chart (unbalanced)
# ------------------------------
```

```
# Refer to Table 7.4 of Montgomery (2013).
x = c(12, 8, 6, 9, 10, 12, 11, 16, 10, 6, 20, 15, 9, 8, 6, 8, 10, 7, 5, 8, 5, 8, 10, 6, 9)
n = c(100,80,80,100,110,110,100,100,90,90,110,120,120,120,110,80,80,80,90,100,100,100,100,90,90)

# The conventional p chart with the unbalanced samples.
# Print LCL, CL, UCL.
# If nk is missing, the average sample size is used.
result = acc(x, n)
print(result)

# Summary of a control chart.
summary(result)

# Plot of a control chart.
# Refer to Figure 7.8 of Montgomery (2013).
plot(result, cex.text=0.8)
text(15, 0.2, labels="p chart (with unbalanced sample)" )


# ================================
# Example 2c: p chart (unbalanced)
#             p is known
# --------------------------------
# Refer to Table 41 of ASTM (2010).
x = c(2, 2, 1, 1, 5, 2, 0, 3, 0, 15, 7, 2, 5, 2, 0, 3, 0, 4, 8, 4)
n = c(600,1300,2000,2500,1550,2000,1550,780,260,2000,1550,950,950,950,35,330,200,600,1300,780)

# The fraction nonconforming is known as 0.0020
# The control limits at the size nk=600.
# If nk (sample size for Phase II) is unknown, the average of subsample sizes is used.
result = acc(x, n, parameter=0.002, nk=600)
summary(result)


# ================================
# Example 3a: u chart (balanced)
# --------------------------------
# Refer to Table 33 of ASTM (2010).
x = c(17, 14, 6, 23,  5,  7, 10, 19, 29, 18, 25, 5,
       8, 11, 18, 13, 22, 6, 23, 22,  9, 15, 20, 6, 24)
n = 10

# The u chart with the balanced samples.
# Print LCL, CL, UCL.
result = acc(x, n, type="u")
print(result)

# Summary of a control chart
summary(result)

# Plot of a control chart
plot(result, cex.text=0.8)
text(13, 3, labels="u chart" )
```

```
# ================================
# Example 3b: u chart (unbalanced)
# --------------------------------
# Refer to Table 34 of ASTM (2010).
x = c(72, 38, 76, 35, 62, 81, 97, 78, 103, 56,
      47, 55, 49, 62, 71, 47, 41, 52, 128, 84)
n = c(20, 20, 40, 25, 25, 25, 40, 40, 40, 40,
      25, 25, 25, 25, 25, 20, 20, 20, 40, 40)

# The u chart with the unbalanced samples.
# Print LCL, CL, UCL.
result = acc(x, n, type="u", nk=20)
print(result)

# Summary of a control chart
summary(result)

# Plot of a control chart
plot(result, cex.text=0.8)
text(12, 3.5, labels="u chart (with unbalanced sample)" )


# ================================
# Example 4: c chart
# --------------------------------
# Refer to Table 35 of ASTM (2010).
x = c(0, 1, 1, 0, 2, 1, 3, 4, 5, 3, 0, 1, 1, 1, 2, 4, 0, 1, 1, 0,
      6, 4, 3, 2, 0, 0, 9,10, 8, 8, 6,14, 0, 1, 2, 4, 5, 7, 1, 3,
      3, 2, 0, 1, 5, 3, 4, 3, 5, 4, 2, 0, 1, 2, 5, 9, 4, 2, 5, 3)

# Print LCL, CL, UCL.
result = acc(x, type="c")
print(result)

# Summary of a control chart
summary(result)

# Plot of a control chart
plot(result, cex.text=0.8)
text(40, 14, labels="c chart" )


# ================================
# Example 5: g and h charts
# --------------------------------
# Refer to Kaminsky et al. (1992).
tmp = c(
11,  2,  8,  2, 4,   1,  1, 11,  2, 1,   1,  7,  1,  1, 9,
 5,  1,  3,  6, 5,  13,  2,  3,  3, 4,   3,  2,  6,  1, 5,
 2,  2,  8,  3, 1,   1,  3,  4,  6, 5,   2,  8,  1,  1, 4,
13, 10, 15,  5, 2,   3,  6,  1,  5, 8,   9,  1, 18,  3, 1,
```

```
 3,  7, 14,  3, 1,   7,  7,  1,  8, 1,   4,  1,  6,  1, 1,
 1, 14,  2,  3, 7,  19,  9,  7,  1, 8,   5,  1,  1,  6, 1,
 9,  5,  6,  2, 2,   8, 15,  2,  3, 3,   4,  7, 11,  4, 6,
 7,  5,  1, 14, 8,   3,  3,  5, 21,10,  11,  1,  6,  1, 2,
 4,  1,  2, 11, 5,   3,  5,  4, 10, 3,   1,  4,  7,  3, 2,
 3,  5,  4,  2, 3,   5,  1,  4, 11,17,   1, 13, 13,  2, 1)
data = matrix(tmp, byrow=TRUE, ncol=5)

# g chart with ML method.
# Print LCL, CL, UCL.
result = acc(data, type="g", location=1)
print(result)

# Summary of a control chart
summary(result)

plot(result, cex.text=0.8)

# h chart with MVU method.
acc(data, type="h", location=1, gEstimator="MVU")


# ==============================
# Example 6: g and h charts (unbalanced data)
# ------------------------------
x1 = c(11, 2,  8,  2, 4)
x2 = c(1,  1, 11,  2, 1)
x3 = c(1,  7,  1)
x4 = c(5,  1,  3,  6, 5)
x5 = c(13, 2,  3,  3)
x6 = c(3,  2,  6,  1, 5)
x7 = c(2,  2,  8,  3, 1)
x8 = c(1,  3,  4,  6, 5)
x9 = c(2,  8,  1,  1, 4)
data = list(x1, x2, x3, x4, x5, x6, x7, x8, x9)

result = acc(data, type="g", location=1, gEstimator="MVU", nk=5)
summary(result)
plot(result)


# ==============================
# Example 7: t charts
# ------------------------------
x = c(0.35, 0.92, 0.59, 4.28, 0.21, 0.79, 1.75, 0.07, 3.3,
1.7, 0.33, 0.97, 0.96, 2.23, 0.88, 0.37, 1.3, 0.4, 0.19, 1.59)

# Exponential t chart
result = acc(x, type="t", tModel="E")
summary(result)

plot(result, cex.text=0.8)
text(10, 6, labels="Exponential t chart" )
```

```
# Weibull t chart
result = acc(x, type="t", tModel="W")
summary(result)

plot(result, cex.text=0.8)
text(10, 6, labels="Weibull t chart" )
```

---

empirical.variance   *Empirical variances of robust estimators*

Description

This function calculate or estimate the variances of the mean, median, Hodges-Lehmann (HL1, HL2, HL3), standard deviation, range, median absolute deviation (MAD) and Shamos estimators.

Usage

```
evar (n, estimator=c("mean","median","HL1","HL2","HL3", "sd","range","mad","shamos"),
        poolType=c("A","B","C"), correction=TRUE )
```

Arguments

n	a vector of sample sizes. For "HL1", "sd", "range" and "shamos", $n \geq 2$. For others, $n \geq 2$.
estimator	a character string specifying the estimator, must be one of "mean" (default), "median", "HL1", "HL2", "HL3", "sd", "range", "mad", and "shamos".
poolType	Type for how to pool estimators, must be one of "A" (default), "B", and "C".
correction	logical. A finite-sample bias correction for the estimator with a single sample. TRUE (default) adjusts a finite-sample bias correction for a scale estimator using c4.factor function.

Details

This function calculates or estimates the variance of a specific estimator when a random sample is from the standard normal distribution.

For the mean, standard deviation (sd) and range, their exact variances are calculated, but the others are empirically estimated through the extensive Monte Carlo simulation with 1E07 replicates for $n = 1, 2, \ldots, 100$. For the case of $n > 100$, the empirical variances are obtained using the method of Hayes (2014).

Value

It returns a numeric value.

Author(s)

Chanseok Park and Min Wang

References

Park, C., H. Kim, and M. Wang (2022). Investigation of finite-sample properties of robust location and scale estimators. *Communications in Statistics - Simulation and Computation*, **51**, 2619-2645. doi:10.1080/03610918.2019.1699114

Hayes, K. (2014). Finite-sample bias-correction factors for the median absolute deviation. *Communications in Statistics: Simulation and Computation*, **43**, 2205–2212.

See Also

RE{rQCC} for the relative efficiency.
n.times.eVar.of.HL1{rQCC} for the empirical variance of the HL1 estimator (times n).
n.times.eVar.of.HL2{rQCC} for the empirical variance of the HL2 estimator (times n).
n.times.eVar.of.HL3{rQCC} for the empirical variance of the HL3 estimator (times n).
n.times.eVar.of.mad{rQCC} for the empirical variance of the MAD estimator (times n).
n.times.eVar.of.median{rQCC} for the empirical variance of the median estimator (times n).
n.times.eVar.of.shamos{rQCC} for the empirical variance of the Shamos estimator (times n).

Examples

```
# Empirical variance of the Hodges-Lehmann estimator (HL2) under the standard normal distribution.
evar (n=10, estimator="HL2")

# Multiple samples
evar (n=c(4,5), estimator="mad", poolType="C")
```

---

factors.for.chart   *Factors for constructing control charts*

Description

Factors for constructing control charts.

Usage

```
factors.cc(n, factor=c( "A","A1","A2","A3", "B1","B2","B3","B4","B5","B6", "c2","c4",
                "d2","d3","D1","D2","D3","D4", "E1", "E2", "E3"), sigmaFactor=3)
```

Arguments

n   sample size ($n \geq 1$).

factor	a character string specifying the foctor.
	"c2", "c4", "d2", "d3" for control chart lines; "A", "A1", "A2", "A3" for averages for computing control limits; "B1", "B2", "B3", "B4", "B5", "B6" for standard devations; "D1", "D2", "D3", "D4" for ranges; and "E1", "E2", "E3" for individuals. For "d3", the calculation is not accurate when $n > 100$.
sigmaFactor	a factor for the standard deviation (σ).
	For example, the American Standard uses "*3*sigma*" limits (0.27% false alarm rate), while the British Standard uses "*3.09*sigma*" limits (0.20% false alarm rate).

Details

The values of the factors are used for constructing various control charts.

For example, the conventional $\bar{X}$ chart with the sample standard deviation is given by

$$\bar{\bar{X}} \pm A_3 \bar{S}.$$

For more details, refer to `vignette("factors.cc", package="rQCC")`.

Value

It returns a numeric value.

Author(s)

Chanseok Park

References

ASTM (2010). *Manual on Presentation of Data and Control Chart Analysis* (STP 15-D), 8th edition. American Society for Testing and Materials, West Conshohocken, PA.

ASTM (1951). *Manual on Quality Control of Materials* (STP 15-C), American Society for Testing and Materials, Philadelphia, PA.

See Also

`c4.factor{rQCC}` for c_4 factor for the finite-sample unbiasing factor to estimate the standard deviation (σ) under the normal distribution using various estimators such as the sample standard deviation, the sample range, the median absolute deviation (MAD), and the Shamos estimator.

Examples

```
## A3 is used for constructing the conventional X-bar chart
#  with the sample standard deviation.
factors.cc(n=10, factor="A3")


## Unbiasing factor for the standard deviation
#         using the sample standard deviation.
factors.cc(n=10, factor="c4")
```

```
# The above is the same as below:
c4.factor(n=10, estimator="sd")


## Unbiasing factor for the standard deviation
#            using the sample range.
factors.cc(n=10, factor="d2")
# The above is the same as below:
c4.factor(n=10, estimator="range")


## Table B2 in Supplement B of ASTM (1951).
char = c("A","A1","A2","c2", "B1","B2","B3","B4", "d2","d3","D1","D2","D3","D4")
nn = 2L:25L

res=NULL
for(n in nn){tmp=NULL;for(ch in char) tmp=c(tmp,factors.cc(n,ch));res=rbind(res,tmp)}
rownames(res) = paste0("n=",nn)
round(res,4)


## Table 49 in Chapter 3 of ASTM (2010).
char = c("A","A2","A3","c4", "B3","B4","B5","B6", "d2","d3","D1","D2","D3","D4")
nn = 2L:25L

res=NULL
for(n in nn){tmp=NULL;for(ch in char) tmp=c(tmp,factors.cc(n,ch));res=rbind(res,tmp)}
rownames(res) = paste0("n=",nn)
round(res,4)


## Table 50 in Chapter 3 of ASTM (2010).
char = c("E2", "E3")
nn = 2L:25L

res=NULL
for(n in nn){tmp=NULL;for(ch in char) tmp=c(tmp,factors.cc(n,ch));res=rbind(res,tmp)}
rownames(res) = paste0("n=",nn)
round(res,3)
```

---

finite.sample.breakdown

Finite-sample breakdown point

Description

Calculates the finite-sample breakdown point of the mean, median, Hodges-Lehmann estimators (HL1, HL2, HL3), standard deviation, range, MAD (median absolute deviation) and Shamos estimators. Note that for the case of the mean, standard deviation and range, the finite-sample breakdown points are always zero.

Usage

```
finite.breakdown (n, estimator=c("mean","median","HL1","HL2","HL3",
                        "sd","range","mad","shamos") )
```

Arguments

n   a numeric vector of sample sizes.

estimator   a character string specifying the estimator, must be one of "mean" (default), "median", "HL1", "HL2", "HL3", "sd", "range", "mad", and "shamos".

Details

`finite.breakdown` gives the finite-sample breakdown point of the specified estimator.

The Hodges-Lehmann (HL1) is defined as

$$\text{HL1} = \operatorname*{median}_{i<j}\left(\frac{X_i + X_j}{2}\right)$$

where $i, j = 1, 2, \ldots, n$.

The Hodges-Lehmann (HL2) is defined as

$$\text{HL2} = \operatorname*{median}_{i \leq j}\left(\frac{X_i + X_j}{2}\right).$$

The Hodges-Lehmann (HL3) is defined as

$$\text{HL3} = \operatorname*{median}_{\forall(i,j)}\left(\frac{X_i + X_j}{2}\right).$$

Value

It returns a numeric value.

Author(s)

Chanseok Park and Min Wang

References

Park, C., H. Kim, and M. Wang (2022). Investigation of finite-sample properties of robust location and scale estimators. *Communications in Statistics - Simulation and Computation*, **51**, 2619-2645. doi:10.1080/03610918.2019.1699114

Hodges, Jr., J. L. (1967). Efficiency in normal samples and tolerance of extreme values for some estimators of location. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. **1**, 163–186. University of California Press, Berkeley.

Hampel, F. R., Ronchetti, E., Rousseeuw, P. J., and Stahel, W. A. (1986). *Robust Statistics: The Approach Based on Influence Functions*, Subsection 2.2a. John Wiley & Sons, New York.

See Also

HL{rQCC} for the Hodges-Lehmann estimator.

Examples

```
# finite-sample breakdown point of the Hodges-Lehmann (HL1) with size n=10.
finite.breakdown(n=10, estimator="HL2")

# finite-sample breakdown points of the median with sizes n=4,5,6
finite.breakdown(n=4:6, estimator="median")
```

Hodges-Lehmann	*Hodges-Lehmann estimator*

Description

Calculates the Hodges-Lehmann estimator.

Usage

```
HL(x, estimator = c("HL1", "HL2", "HL3"), na.rm = FALSE)
```

Arguments

x	a numeric vector of observations.
estimator	a character string specifying the estimator, must be one of "HL1" (default), "HL2" and "HL3".
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.

Details

HL computes the Hodges-Lehmann estimators (one of "HL1", "HL2", "HL3").

The Hodges-Lehmann (HL1) is defined as

$$\text{HL1} = \operatorname*{median}_{i<j} \left(\frac{X_i + X_j}{2} \right)$$

where $i, j = 1, 2, \ldots, n$.

The Hodges-Lehmann (HL2) is defined as

$$\text{HL2} = \operatorname*{median}_{i \leq j} \left(\frac{X_i + X_j}{2} \right).$$

The Hodges-Lehmann (HL3) is defined as

$$\text{HL3} = \operatorname*{median}_{\forall(i,j)} \left(\frac{X_i + X_j}{2} \right).$$

Value

It returns a numeric value.

Author(s)

Chanseok Park and Min Wang

References

Park, C., H. Kim, and M. Wang (2022). Investigation of finite-sample properties of robust location and scale estimators. *Communications in Statistics - Simulation and Computation*, **51**, 2619-2645. doi:10.1080/03610918.2019.1699114

Hodges, J. L. and E. L. Lehmann (1963). Estimates of location based on rank tests. *Annals of Mathematical Statistics*, **34**, 598–611.

See Also

mean{base} for calculating sample mean and median{stats} for calculating sample median.

`finite.breakdown`{rQCC} for calculating the finite-sample breakdown point.

Examples

```
x = c(0:10, 50)
HL(x, estimator="HL2")
```

MAD   *Median absolute deviation (MAD)*

Description

Calculates the unbiased median absolute deviation (MAD) estimator and the unbiased squared MAD under the normal distribution which are adjusted by the Fisher-consistency and finite-sample unbiasing factors.

Usage

```
mad.unbiased (x, center = median(x), constant=1.4826, na.rm = FALSE)

mad2.unbiased(x, center = median(x), constant=1.4826, na.rm = FALSE)
```

Arguments

x	a numeric vector of observations.
center	pptionally, the center: defaults to the median.
constant	correction factor for the Fisher-consistency under the normal distribution
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.

Details

The unbiased MAD (mad.unbiased) is defined as the mad{stats} divided by $c_5(n)$, where $c_5(n)$ is the finite-sample unbiasing factor. Note that $c_5(n)$ notation is used in Park et. al (2022), and $c_5(n)$ is calculated using the function c4.factor{rQCC} with estimator="mad" option. The default value (constant=1.4826) ensures the Fisher-consistency under the normal distribution. Note that the original MAD was proposed by Hampel (1974).

The unbiased squared MAD (mad2.unbiased) is defined as the squared mad{stats} divided by $w_5(n)$ where $w_5(n)$ is the finite-sample unbiasing factor. Note that $w_5(n)$ notation is used in Park et. al (2022), and $w_5(n)$ is calculated using the function w4.factor{rQCC} with estimator="mad2" option. The default value (constant=1.4826) ensures the Fisher-consistency under the normal distribution. Note that the square of the conventional MAD is Fisher-consistent for the variance (σ^2) under the normal distribution, but it is not unbiased with a sample of finite size.

Value

They return a numeric value.

Author(s)

Chanseok Park and Min Wang

References

Park, C., H. Kim, and M. Wang (2022). Investigation of finite-sample properties of robust location and scale estimators. *Communications in Statistics - Simulation and Computation*, **51**, 2619-2645. doi:10.1080/03610918.2019.1699114

Hampel, F. R. (1974). The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, **69**, 383–393.

See Also

c4.factor{rQCC} for finite-sample unbiasing factor for the standard deviation under the normal distribution.

w4.factor{rQCC} for finite-sample unbiasing factor for the variance under the normal distribution.

shamos{rQCC} for robust Fisher-consistent estimator of the standard deviation under the normal distribution.

shamos.unbiased{rQCC} for robust finite-sample unbiased estimator of the standard deviation under the normal distribution.

mad{stats} for calculating the sample MAD.

finite.breakdown{rQCC} for calculating the finite-sample breakdown point.

Examples

```
x = c(0:10, 50)

# Fisher-consistent MAD, but not unbiased with a finite sample.
mad(x)
```

```
# Unbiased MAD.
mad.unbiased(x)

# Fisher-consistent squared MAD, but not unbiased.
mad(x)^2

# Unbiased squared MAD.
mad2.unbiased(x)
```

---

n.times.eBias   *Empirical biases (times n)*

Description

n times the empirical biases of the median absolute deviation (MAD) and Shamos estimators.

Usage

```
n.times.eBias.of.mad

n.times.eBias.of.shamos
```

Details

n times the empirical biases of the median absolute deviation (MAD) and Shamos estimators under the standard normal distribution, where n is the sample size and n is from 1 to 100. For the MAD estimator, mad{stats} is used. For the Shamos estimator, the Fisher-consistent Shamos estimator, shamos{rQCC}, is used.

These estimators are not unbiased with a finite sample. The empirical biases are obtained using the extensive Monte Carlo simulation with 1E07 replicates.

Value

They return a vector of 100 values.

Author(s)

Chanseok Park and Min Wang

References

Park, C., H. Kim, and M. Wang (2022). Investigation of finite-sample properties of robust location and scale estimators. *Communications in Statistics - Simulation and Computation*, **51**, 2619-2645. doi:10.1080/03610918.2019.1699114

Shamos, M. I. (1976). Geometry and statistics: Problems at the interface. In Traub, J. F., editor, *Algorithms and Complexity: New Directions and Recent Results*, pages 251–280. Academic Press, New York.

Lèvy-Leduc, C., Boistard, H., Moulines, E., Taqqu, M. S., and Reisen, V. A. (2011). Large sample behaviour of some well-known robust estimators under long-range dependence. *Statistics*, **45**, 59–71.

---

n.times.eVar *Empirical variances (times n)*

Description

n times the empirical variances of the Hodges-Lehmann (HL1, HL2, HL3), the median, the median absolute deviation (MAD), and the Shamos estimators.

Usage

```
n.times.eVar.of.HL1

n.times.eVar.of.HL2

n.times.eVar.of.HL3

n.times.eVar.of.median

n.times.eVar.of.mad

n.times.eVar.of.shamos
```

Details

n times the empirical variances of the Hodges-Lehmann (HL1, HL2, HL3), the median, the median absolute deviation (MAD), and the Shamos estimators under the standard normal distribution, where n is the sample size and n is from 1 to 100.

For the MAD estimators, mad{stats} is used. For the Hodges-Lehmann, HL{rQCC} is used. For the Shamos, the Fisher-consistent Shamos estimator, shamos{rQCC}, is used.

The empirical variances are obtained using the Monte Carlo simulation with 1E07 replicates.

Value

They return a vector of 100 values.

Author(s)

Chanseok Park and Min Wang

References

Park, C., H. Kim, and M. Wang (2022). Investigation of finite-sample properties of robust location and scale estimators. *Communications in Statistics - Simulation and Computation*, **51**, 2619-2645. doi:10.1080/03610918.2019.1699114

Hodges, J. L. and E. L. Lehmann (1963). Estimates of location based on rank tests. *Annals of Mathematical Statistics*, **34**, 598–611.

Shamos, M. I. (1976). Geometry and statistics: Problems at the interface. In Traub, J. F., editor, *Algorithms and Complexity: New Directions and Recent Results*, pages 251–280. Academic Press, New York.

Lèvy-Leduc, C., Boistard, H., Moulines, E., Taqqu, M. S., and Reisen, V. A. (2011). Large sample behaviour of some well-known robust estimators under long-range dependence. *Statistics*, **45**, 59–71.

---

plot   *Plot quality control chart*

Description

Plot method for an object of class "acc", "rcc", and "racc".

Usage

```
## S3 method for class 'acc'
plot(x, digits= getOption("digits")-2, col.line="cyan4",
                 col.background="ivory", col.boundary="ivory4",
                 cex.text=0.7, x.text, text.offset=0, LCL, CL, UCL, ...)

## S3 method for class 'rcc'
plot(x, digits= getOption("digits")-2, col.line="cyan4",
                 col.background="ivory", col.boundary="ivory4",
                 cex.text=0.7, x.text, text.offset=0, LCL, CL, UCL, ...)

## S3 method for class 'racc'
plot(x, digits= getOption("digits")-2, col.line="cyan4",
                 col.background="ivory", col.boundary="ivory4",
                 cex.text=0.7, x.text, text.offset=0, LCL, CL, UCL, ...)
```

Arguments

x	an object of class "rcc".
digits	number of significant digits to use, see print.
col.line	line color connecting subgroup estimates.
col.background	background color for in-control region.
col.boundary	boundary color for in-control region.

cex.text	magnification to be used for the text labels (LCL, CL, UCL).
x.text	a vector of x-coordinates where the text labels (LCL, CL, UCL) should be written. If missing, the text labels will be written on the right side.
text.offset	an offset value for the text labes (LCL, CL, UCL).
LCL	lower control limit. If missing, use the value from an object of class "rcc" or "acc".
CL	center line. If missing, use the value from an object of class "rcc" or "acc".
UCL	upper control limit. If missing, use the value from an object of class "rcc" or "acc".
...	additional parameters to plot.

Author(s)

Chanseok Park

See Also

acc{rQCC}, rcc{rQCC}, racc{rQCC}, plot

---

pooled.Estimator	*Pooled Estimator*

Description

This function calculates the pooled estimator based on the unbiased estimators such as the mean, median, Hodges-Lehmann (HL1, HL2, HL3), standard deviation, range, median absolute deviation (MAD) and Shamos estimators.

Usage

```
pooledEstimator(x, estimator = c("mean", "median", "HL1", "HL2", "HL3",
                          "sd", "range", "mad", "shamos"),
              poolType=c("A", "B", "C") )
```

Arguments

x	a numeric list of observations.
estimator	a character string specifying the estimator, must be one of "mean" (default), "median", "HL1", "HL2", "HL3", "sd", "range", "mad", and "shamos".
poolType	Type for how to pool estimators, must be one of "A" (default), "B", and "C".

Details

This function calculates the pooled estimator based on one of "mean" (default), "median", "HL1", "HL2", "HL3", "sd", "mad", and "shamos", which are all unbiased. There are three different methods of pooling the estimators, denoted by "A" (default), "B", and "C". For more details on how to pool them, refer to vignette.

Value

They return a numeric value.

Author(s)

Chanseok Park

References

Park, C. and M. Wang (2020). A study on the X-bar and S control charts with unequal sample sizes. *Mathematics*, **8**(5), 698.
doi:10.3390/math8050698

Park, C., H. Kim, and M. Wang (2022). Investigation of finite-sample properties of robust location and scale estimators. *Communications in Statistics - Simulation and Computation*, **51**, 2619-2645.
doi:10.1080/03610918.2019.1699114

Examples

```
x1 = c(1,2,3,4,5)
x2 = c(6,7)
x  = list(x1,x2)

# Pooled sample mean (default) by type "A" pooling
pooledEstimator(x)
pooledEstimator(x, "mean", "A") # same as the above

# Pooled sample mean by type "B" pooling
pooledEstimator(x, "mean", "B")


# Pooled sample sd by type "B" pooling
pooledEstimator(x, estimator="sd", pool="B")
```

---

relative.efficiency   *Relative efficiency (RE)*

Description

RE calculates the relative efficiency value of a location estimator (″median″, ″HL1″, ″HL2″, ″HL3″) with respect to the sample mean and that of a scale estimator (″range″, ″mad″, ″shamos″) with respect to the sample standard deviation.

Usage

```
RE(n, estimator=c("mean", "median", "HL1", "HL2", "HL3", "sd", "range", "mad", "shamos"),
      poolType =c("A", "B", "C"),
      baseEstimator, basePoolType, correction=TRUE, correctionBase)
```

Arguments

n	a vector of sample sizes ($n_i \geq 1$).
estimator	a character string specifying the estimator, must be one of "mean" (default), "median", "HL1", "HL2", "HL3", "sd", "range", "mad", and "shamos".
poolType	Type for how to pool estimators, must be one of "A" (default), "B", and "C".
baseEstimator	a character string specifying the baseline estimator on the numerator of the relative efficiency, must be one of "mean" (default), "median", "HL1", "HL2", "HL3", "sd", "range", "mad", and "shamos".
basePoolType	Type for how to pool baseline estimator, must be one of "A", "B", and "C". If missing, basePoolType <- poolType is used.
correction	logical. A finite-sample bias correction for the estimator with a single sample. TRUE (default) adjusts a finite-sample bias correction for a scale estimator using c4.factor function.
correctionBase	logical. A finite-sample bias correction for the baseline estimator with a single sample. If missing, correctionBase <- correction is used.

Details

Under the assumption of the normal distribution, the function calculates the relative efficiency value of the mean, median and Hodges-Lehmann (HL1, HL2, HL3) estimators with respect to the selected baseline estimator (default is the sample mean) and that of the standard deviation, range, median absolute deviation (MAD) and Shamos estimators with respect to the selected baseline estimator (default is the sample standard deviation).

For the case of the sample mean, standard deviation and range, it is possible to derive their variances in analytic form, but, for the other case, it may be impossible. In this case, the variances with $n = 1, 2, \ldots, 100$ are obtained using the extensive Monte Carlo simulation with 1E07 replicates. For $n > 100$, the variances are approximated based on the method of Hayes (2014).

To obtain the relative efficiency value of the unbiased scale estimators, use correction=TRUE option. Note that the location estimators ("mean", "median", "HL1", "HL2", "HL3") are unbiased.

If n is a vector of multiple values (multiple samples), the RE function calculates the relative efficiency value of the pooled estimator. In this case, only unbiased pooled estimator and baseline estimator are considered. That is, we use correction=TURE and correctionBase=TURE for multiple samples.

Note that the relative efficiency (RE) of $\hat{\theta}_2$ with respect to $\hat{\theta}_1$ is defined as

$$\mathrm{RE}(\hat{\theta}_2|\hat{\theta}_1) = \frac{\mathrm{Var}(\hat{\theta}_1)}{\mathrm{Var}(\hat{\theta}_2)}.$$

Value

It returns a numeric value.

Author(s)

Chanseok Park

References

Park, C., H. Kim, and M. Wang (2022). Investigation of finite-sample properties of robust location and scale estimators. *Communications in Statistics - Simulation and Computation*, **51**, 2619-2645. doi:10.1080/03610918.2019.1699114

Hayes, K. (2014). Finite-sample bias-correction factors for the median absolute deviation. *Communications in Statistics: Simulation and Computation*, **43**, 2205–2212.

See Also

n.times.eVar.of.HL1{rQCC} for the empirical variance of the HL1 estimator (times n).
n.times.eVar.of.HL2{rQCC} for the empirical variance of the HL2 estimator (times n).
n.times.eVar.of.HL3{rQCC} for the empirical variance of the HL3 estimator (times n).
n.times.eVar.of.mad{rQCC} for the empirical variance of the MAD estimator (times n).
n.times.eVar.of.median{rQCC} for the empirical variance of the median estimator (times n).
n.times.eVar.of.shamos{rQCC} for the empirical variance of the Shamos estimator (times n).

Examples

```
#################
# Single sample #
#################


# RE of the Hodges-Lehmann (HL2) estimator
#  with respect to the sample standard deviation under the normal distribution.
RE(n=5, estimator="HL2")


# RE of the unbiased Shamos estimator
# with respect to the unbiased sample standard deviation under the normal distribution.
RE(n=5, estimator="shamos")


# RE of the original Shamos estimator
# with respect to the sample standard deviation under the normal distribution.
RE(n=5, estimator="shamos", correction=FALSE)


# RE of the unbiased range ( (maximum - minimum) / d2 )
# with respect to the unibased sample standard deviation under the normal distribution.
RE(n=6, estimator="range")


# RE of the original range (maximum minus minimum)
# with respect to the sample standard deviation under the normal distribution.
RE(n=6, estimator="range", correction=FALSE)



####################
# Multiple samples #
####################
# With multiple samples, only the unbiased pooled estimators are considered.


# RE of the pooled median (pooling type A) with respect to the mean (pooling type A)
RE( n=c(4,5), estimator="median" )
```

```
# RE of the pooled median (pooling type A) with respect to the median (pooling type C)
RE( n=c(4,5), estimator="median", baseEstimator="median", basePoolType="C")

# RE of the pooled mad (pooling type A) with respect to the standard deviation (pooling type A)
RE( n=c(4,5), estimator="mad")

# RE of the pooled mad (pooling type A) with respect to the standard deviation (pooling type C)
RE( n=c(4,5), estimator="mad", basePoolType="C")

# RE of the pooled standard deviation (pooling type A) with respect to the sd (pooling type C)
RE( n=c(4,5), estimator="sd", baseEstimator="sd", basePoolType="C" )
```

---

robust.attributes.chart.unbalanced

Robust attributes control charts with balanced/unbalanced samples

---

Description

Constructs the robust g and h attributes control charts with balanced/unbalanced samples.

Usage

```
racc (x, gamma, type=c("g","h","t"), parameter, gEstimator=c("cdf", "MM"),
        tModel=c("E", "W"), location.shift = 0, sigmaFactor=3, nk)
```

Arguments

x	a numeric vector of the number of nonconforming units.
gamma	a numeric value for a inlier proportion. gamma should be between 0 and 1 (smaller value means more trimming).
type	a character string specifying the type of control chart.
parameter	a known Bernoulli parameter value for the g and h charts. If not known, it is estimated. For more details, refer to vignette("racc", package="rQCC").
gEstimator	a method for estimating the Bernoulli parameter for g and h charts. "cdf" is based on the memoryless property and "MM" is based on the truncated geometric distribution.
tModel	Probability model for t chart. "E" for Exponential and "W" for Weibull.
location.shift	a known location shift parameter value for g and h charts.
sigmaFactor	a factor for the standard deviation (σ). For example, the American Standard uses "*3*sigma*" limits (0.27% false alarm rate), while the British Standard uses "*3.09*sigma*" limits (0.20% false alarm rate).
nk	sample size for Phase-II. If nk is missing, the average of the subsample sizes is used.

Details

racc constructs the attributes control charts for nonconforming units (p and np charts) and for nonconformities per unit (c and u charts).

Value

racc returns an object of class "racc". The function summary is used to obtain and print a summary of the results and the function plot is used to plot the control chart.

Author(s)

Chanseok Park

References

Park, C., L. Ouyang, and M. Wang (2021). Robust g-type quality control charts for monitoring nonconformities. *Computers and Industrial Engineering*, **162**, 107765.

Kaminsky, F. C., J. C. Benneyan and R. D. Davis (1992). Statistical Control Charts Based on a Geometric Distribution. *Journal of Quality Technology*, **24**, 63-69.

Examples

```
# ==============================
# Example 1: g and h charts
# ------------------------------
# Refer to Kaminsky et al. (1992) and Table 2 of Park, et al. (2021).
tmp = c(
11,  2,  8,  2, 4,   1,  1, 11,  2, 1,   1,  7,  1,  1, 9,
 5,  1,  3,  6, 5,  13,  2,  3,  3, 4,   3,  2,  6,  1, 5,
 2,  2,  8,  3, 1,   1,  3,  4,  6, 5,   2,  8,  1,  1, 4,
13, 10, 15,  5, 2,   3,  6,  1,  5, 8,   9,  1, 18,  3, 1,
 3,  7, 14,  3, 1,   7,  7,  1,  8, 1,   4,  1,  6,  1, 1,
 1, 14,  2,  3, 7,  19,  9,  7,  1, 8,   5,  1,  1,  6, 1,
 9,  5,  6,  2, 2,   8, 15,  2,  3, 3,   4,  7, 11,  4, 6,
 7,  5,  1, 14, 8,   3,  3,  5, 21,10,  11,  1,  6,  1, 2,
 4,  1,  2, 11, 5,   3,  5,  4, 10, 3,   1,  4,  7,  3, 2,
 3,  5,  4,  2, 3,   5,  1,  4, 11,17,   1, 13, 13,  2, 1)
data = matrix(tmp, byrow=TRUE, ncol=5)

# g chart with cdf (trimming) method.
# Print LCL, CL, UCL.
result = racc(data, gamma=0.9, type="g", location=1)
print(result)

# Summary of a control chart
summary(result)

plot(result, cex.text=0.8)

# h chart with MM (truncated geometric) method.
racc(data, gamma=0.9, type="h", location=1, gEstimator="MM")
```

```
# ==============================
# Example 2: g and h charts (unbalanced data)
# ------------------------------
x1 = c(11, 2,  8,  2, 4)
x2 = c(1,  1, 11,  2, 1)
x3 = c(1,  7,  1)
x4 = c(5,  1,  3,  6, 5)
x5 = c(13, 2,  3,  3)
x6 = c(3,  2,  6,  1, 5)
x7 = c(2,  2,  8,  3, 1)
x8 = c(1,  3,  4,  6, 5)
x9 = c(2,  8,  1,  1, 4)
data = list(x1, x2, x3, x4, x5, x6, x7, x8, x9)

result = racc(data, gamma=0.9, type="g", location=1, gEstimator="cdf", nk=5)
summary(result)
plot(result)


# ==============================
# Example 3: t charts
# ------------------------------
x = c(0.35, 0.92, 0.59, 4.28, 0.21, 0.79, 1.75, 0.07, 3.3,
1.7, 0.33, 0.97, 0.96, 2.23, 0.88, 0.37, 1.3, 0.4, 0.19, 1.59)

# Exponential t chart
result = racc(x, type="t", tModel="E")
summary(result)

plot(result, cex.text=0.8)
text(10, 6, labels="Robust exponential t chart" )


# Weibull t chart
result = racc(x, type="t", tModel="W")
summary(result)

plot(result, cex.text=0.8)
text(10, 5.5, labels="Robust Weibull t chart" )
```

```
robust.chart.unbalanced
```
                    Robust quality control chart with balanced/unbalanced samples

---

Description

Constructs the robust control charts with balanced/unbalanced samples.

Usage

```
rcc (x, location = c("mean", "median", "HL1", "HL2", "HL3"),
     scale = c("sd", "range", "mad", "shamos"), type = c("Xbar", "S", "R"),
     poolLoc = c("A", "B", "C"), poolScale = c("A", "B", "C"), sigmaFactor=3, nk)
```

Arguments

x	a numeric matrix or list of vectors. Each row or vector contains a sub-sample.
location	a character string specifying the location estimator, must be one of "mean" (default), "median", "HL1", "HL2" and "HL3".
scale	a character string specifying the scale estimator, must be one of "sd" (default), "range", "mad" and "shamos".
type	a character string specifying the type of control chart.
poolLoc	pooling type for location.
poolScale	pooling type for scale.
sigmaFactor	a factor for the standard deviation (σ). For example, the American Standard uses "*3*sigma*" limits (0.27% false alarm rate), while the British Standard uses "*3.09*sigma*" limits (0.20% false alarm rate).
nk	sample size for Phase-II. If nk is missing, the average sample size is used.

Details

rcc constructs the robust X-bar, S and R control charts. Using various robust location and scale estimators, one can construct the robust control charts. For more details on the control charts, refer to vignette("rcc", package="rQCC").

Note that the location and scale estimators used in this fuction are all unbiased. For more details on how to pool the location and scale estimators, refer to vignette("pooledEstimator", package="rQCC").

In addition, one can also construct the conventional $\bar{X}$ chart with S and $\bar{X}$ chart with R.

Value

rcc returns an object of [class](#) "rcc". The function summary is used to obtain and print a summary of the results and the function plot is used to plot the control chart.

Author(s)

Chanseok Park and Min Wang

References

Park, C., L. Ouyang, and M. Wang (2022). Development of robust X-bar charts with unequal sample sizes. *ArXiv e-prints*, 2212.10731.
[doi:10.48550/arXiv.2212.10731](https://doi.org/10.48550/arXiv.2212.10731)

Park, C., H. Kim, and M. Wang (2022). Investigation of finite-sample properties of robust location and scale estimators. *Communications in Statistics - Simulation and Computation*, **51**, 2619-2645. doi:10.1080/03610918.2019.1699114

ASTM (2010). *Manual on Presentation of Data and Control Chart Analysis* (STP 15-D), 8th edition. American Society for Testing and Materials, West Conshohocken, PA.

Ryan (2000). *Statistical Methods For Quality Improvement*, 2nd edition. John Wiley & Sons, New York, NY.

Montgomery (2013). *Statistical Quality Control: An Modern Introduction*, 7th edition. John Wiley & Sons, New York, NY.

Examples

```
###############
# X-bar chart #
###############


# ==========
# Example 1a
# ----------
# The conventional X-bar chart with the standard deviation.
# Refer to Example 3 in Section 3.31 of ASTM (2010).

# The data below are from Table 29 in Section 3.31 of ASTM (2010).
# Each subgroup has a sample of size n=6. There are m=10 subgroups.
x1 = c(0.5005, 0.5000, 0.5008, 0.5000, 0.5005, 0.5000)
x2 = c(0.4998, 0.4997, 0.4998, 0.4994, 0.4999, 0.4998)
x3 = c(0.4995, 0.4995, 0.4995, 0.4995, 0.4995, 0.4996)
x4 = c(0.4998, 0.5005, 0.5005, 0.5002, 0.5003, 0.5004)
x5 = c(0.5000, 0.5005, 0.5008, 0.5007, 0.5008, 0.5010)
x6 = c(0.5008, 0.5009, 0.5010, 0.5005, 0.5006, 0.5009)
x7 = c(0.5000, 0.5001, 0.5002, 0.4995, 0.4996, 0.4997)
x8 = c(0.4993, 0.4994, 0.4999, 0.4996, 0.4996, 0.4997)
x9 = c(0.4995, 0.4995, 0.4997, 0.4992, 0.4995, 0.4992)
x10= c(0.4994, 0.4998, 0.5000, 0.4990, 0.5000, 0.5000)
data1 = rbind(x1, x2, x3, x4, x5, x6, x7, x8, x9, x10)

# Print LCL, CL, UCL.
# The mean and standard deviation are used.
result = rcc(data1, loc="mean", scale="sd", type="Xbar")
print(result)

# Note: X-bar chart is a default with the mean and sd
#       so the below is the same as the above.
rcc(data1)

# Summary of a control chart
summary(result)
RE(n=6, estimator="sd", correction=TRUE)

# The above limits are also calculated as
A3 = factors.cc(n=6, "A3")
```

```
xbarbar = mean(data1)
 # xbarbar = mean(unlist(data1)) # for list

sbar = mean( apply(data1, 1, sd) )
c(xbarbar-A3*sbar, xbarbar, xbarbar+A3*sbar)

# Plot a control chart
plot(result, cex.text=0.8)
abline(v=5.5, lty=1, lwd=2, col="gold")
text( c(3,8), c(0.5005, 0.5005), labels=c("Group 1", "Group 2") )

# ==========
# Example 1b
# ----------
# The conventional X-bar chart with the range.
# Refer to Example 5 in Section 3.31 of ASTM (2010).
# The data are the same as in Example 1a.

# Print LCL, CL, UCL.
# The range is used for the scale estimator.
result = rcc(data1, loc="mean", scale="range")
print(result)

# Summary of a control chart
# Note: the RE is calculated based on the unbiased estimators.
summary(result)
RE(n=6, estimator="range", correction=TRUE)

# The above limits are also calculated as
A2 = factors.cc(n=6, "A2")

xbarbar = mean(data1)
 # xbarbar = mean(unlist(data1)) # for list

Rbar = mean( apply(data1, 1, function(x) {diff(range(x))}) )
 # Rbar = mean( apply(sapply(data1,range),2,diff) ) # for list

c(xbarbar-A2*Rbar, xbarbar, xbarbar+A2*Rbar)

# Plot a control chart
plot(result, cex.text=0.8)
abline(v=5.5, lty=1, lwd=2, col="gold")
text( c(3,8), c(0.5005, 0.5005), labels=c("Group 1", "Group 2") )

# ==========
# Example 1c
# ----------
# The median-MAD chart.
# Refer to Table 4.2 in Section 4.7 of Ryan (2000).
# Data: 20 subgroups with 4 observations each.
tmp = c(
72, 84, 79, 49, 56, 87, 33, 42, 55, 73, 22, 60, 44, 80, 54, 74,
```

```
        97, 26, 48, 58, 83, 89, 91, 62, 47, 66, 53, 58, 88, 50, 84, 69,
        57, 47, 41, 46, 13, 10, 30, 32, 26, 39, 52, 48, 46, 27, 63, 34,
        49, 62, 78, 87, 71, 63, 82, 55, 71, 58, 69, 70, 67, 69, 70, 94,
        55, 63, 72, 49, 49, 51, 55, 76, 72, 80, 61, 59, 61, 74, 62, 57 )
    data2 = matrix(tmp, ncol=4, byrow=TRUE)

    # Print LCL, CL, UCL.
    # The median (location) and MAD (scale) are used.
    rcc(data2, loc="median", scale="mad")

    # Note: the RE is calculated based on the unbiased estimators.
    RE(n=4, estimator="median", correction=TRUE)

    # ==========
    # Example 1d
    # ----------
    # The HL2-Shamos chart.
    # The data are the same as in Example 1c.

    # Print LCL, CL, UCL.
    # The HL2 (location) and Shamos (scale) are used.
    rcc(data2, loc="HL2", scale="shamos")

    # Note: the RE is calculated based on the unbiased estimators.
    RE(n=4, estimator="HL2", correction=TRUE)


    ############
    # S chart  #
    ############

    # ==========
    # Example 2a
    # ----------
    # The conventional S chart with the standard deviation.
    # Refer to Example 3 in Section 3.31 of ASTM (2010).
    # The data are the same as in Example 1a.

    # Print LCL, CL, UCL.
    # The standard deviaion (default) is used for the scale estimator.
    result = rcc(data1, type="S")
    print(result)

    # The above limits are also calculated as
    B3 = factors.cc(n=6, "B3")
    B4 = factors.cc(n=6, "B4")
    sbar = mean( apply(data1, 1, sd) )
    c(B3*sbar, sbar, B4*sbar)

    # Plot a control chart
    plot(result, cex.text=0.8)
    abline(v=5.5, lty=1, lwd=2, col="gold")
    text( c(3,8), c(0.0005, 0.0005), labels=c("Group 1", "Group 2") )
```

```
# ==========
# Example 2b
# ----------
# The S-type chart with the MAD.
# The data are the same as in Example 2a.

# Print LCL, CL, UCL.
# The mad (scale) are used.
result = rcc(data1, scale="mad", type="S")
print(result)

# Plot a control chart
plot(result, cex.text=0.8)
abline(v=5.5, lty=1, lwd=2, col="gold")
text( c(3,8), c(0.00045, 0.00045), labels=c("Group 1", "Group 2") )


############
# R chart  #
############

# ==========
# Example 3a
# ----------
# The conventional R chart with the range.
# Refer to Example 5 in Section 3.31 of ASTM (2010).
# The data are the same as in Example 1a.

# Print LCL, CL, UCL.
# The range is used for the scale estimator.
# Unlike the S chart, scale="range" is not a default.
# Thus, for the conventional R chart, use the option (scale="range") as below.
result = rcc(data1, scale="range", type="R")
print(result)

# The above limits are also calculated as
D3 = factors.cc(n=6, "D3")
D4 = factors.cc(n=6, "D4")

Rbar = mean( apply(data1, 1, function(x) {diff(range(x))}) )
 # Rbar = mean( apply(sapply(data1,range),2,diff) ) # for list

c(D3*Rbar, Rbar, D4*Rbar)

# Plot a control chart
plot(result, cex.text=0.8)
abline(v=5.5, lty=1, lwd=2, col="gold")
text( c(3,8), c(0.00135, 0.00135), labels=c("Group 1", "Group 2") )


# ==========
```

```
# Example 3b
# ----------
# The R-type chart with the Shamos.
# Refer to Example 5 in Section 3.31 of ASTM (2010).
# The data are the same as in Example 3a.

# Print LCL, CL, UCL.
# The mad (scale) are used.
result = rcc(data1, scale="shamos", type="R")
print(result)

# Plot a control chart
plot(result, cex.text=0.8)
abline(v=5.5, lty=1, lwd=2, col="gold")
text( c(3,8), c(0.00135, 0.00135), labels=c("Group 1", "Group 2") )


####################
# Unbalanced Data #
####################

# ==========
# Example 4
# ----------
# Refer to Example 4 in Section 3.31 of ASTM (2010).
# Data set is from Table 30 in Section 3.31 of ASTM (2010).
 x1 = c( 73, 73, 73, 75, 75)
 x2 = c( 70, 71, 71, 71, 72)
 x3 = c( 74, 74, 74, 74, 75)
 x4 = c( 70, 70, 70, 72, 73)
 x5 = c( 70, 70, 70, 70, 70)
 x6 = c( 65, 65, 66, 69, 70)
 x7 = c( 72, 72, 74, 76)
 x8 = c( 69, 70, 71, 73, 73)
 x9 = c( 71, 71, 71, 71, 72)
x10 = c( 71, 71, 71, 71, 72)
x11 = c( 71, 71, 72, 72, 72)
x12 = c( 70, 71, 71, 72, 72)
x13 = c( 73, 74, 74, 75, 75)
x14 = c( 74, 74, 75, 75, 75)
x15 = c( 72, 72, 72, 73, 73)
x16 = c( 75, 75, 75, 76)
x17 = c( 68, 69, 69, 69, 70)
x18 = c( 71, 71, 72, 72, 73)
x19 = c( 72, 73, 73, 73, 73)
x20 = c( 68, 69, 70, 71, 71)
x21 = c( 69, 69, 69, 69, 69)

# For unbalanced data set, use list.
data = list(x1, x2,  x3, x4,  x5,  x6,  x7,  x8,  x9, x10,
            x11,x12,x13,x14, x15, x16, x17, x18, x19, x20, x21)

# Xbar chart (witn nk=5)
```

```
rcc(data, nk=5)

# S chart (witn nk=4)
rcc(data, type="S", nk=4)


# ==========
# Example 5a
# ----------
# Data set is from Example 6.4 of Montgomery (2013)
#     Statistical Quality Control (7th ed), Wiley.
# Data set for Phase I
 x1 = c(74.030, 74.002, 74.019, 73.992, 74.008)
 x2 = c(73.995, 73.992, 74.001)
 x3 = c(73.988, 74.024, 74.021, 74.005, 74.002)
 x4 = c(74.002, 73.996, 73.993, 74.015, 74.009)
 x5 = c(73.992, 74.007, 74.015, 73.989, 74.014)
 x6 = c(74.009, 73.994, 73.997, 73.985)
 x7 = c(73.995, 74.006, 73.994, 74.000)
 x8 = c(73.985, 74.003, 73.993, 74.015, 73.988)
 x9 = c(74.008, 73.995, 74.009, 74.005)
x10 = c(73.998, 74.000, 73.990, 74.007, 73.995)
x11 = c(73.994, 73.998, 73.994, 73.995, 73.990)
x12 = c(74.004, 74.000, 74.007, 74.000, 73.996)
x13 = c(73.983, 74.002, 73.998)
x14 = c(74.006, 73.967, 73.994, 74.000, 73.984)
x15 = c(74.012, 74.014, 73.998)
x16 = c(74.000, 73.984, 74.005, 73.998, 73.996)
x17 = c(73.994, 74.012, 73.986, 74.005)
x18 = c(74.006, 74.010, 74.018, 74.003, 74.000)
x19 = c(73.984, 74.002, 74.003, 74.005, 73.997)
x20 = c(74.000, 74.010, 74.013)
x21 = c(73.982, 74.001, 74.015, 74.005, 73.996)
x22 = c(74.004, 73.999, 73.990, 74.006, 74.009)
x23 = c(74.010, 73.989, 73.990, 74.009, 74.014)
x24 = c(74.015, 74.008, 73.993, 74.000, 74.010)
x25 = c(73.982, 73.984, 73.995, 74.017, 74.013)

# For unbalanced data set, use list.
data = list(x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12, x13, x14, x15,
            x16, x17, x18, x19, x20, x21, x22, x23, x24, x25)

# Xbar chart (witn nk=5)
rcc(data, nk=5)

# Xbar chart (witn nk=5) with different pooling methods
rcc(data, nk=5, poolLoc="C", poolScale="C")

# S chart (witn nk=5)
rcc(data, type="S", nk=5)

# S chart (witn nk=5) with plling method C.
rcc(data, type="S", nk=5, poolScale="C")
```

```
# ==========
# Example 5b
# ----------
# With contaminated data set.
# Two contaminated observations are added
#   in the first subgroup (70.5, 77.0) of Example 5a.
datan = data
datan[[1]] = c(data[[1]], 70.5, 77.0)

# Xbar chart with non-robust estimators
rcc(datan, nk=5)

# robust Xbar chart (median and mad estimates)
rcc(datan, loc="median", sc="mad", nk=5)

# robust Xbar chart (median and mad estimates) with different pooling methods
rcc(datan, loc="median", sc="mad", nk=5, poolLoc="C", poolScale="C")

# robust S chart (mad estimate) with different pooling methods
rcc(datan, type="S", sc="mad", nk=5, poolScale="B")
rcc(datan, type="S", sc="mad", nk=5, poolScale="C")
```

---

sd.unbiased   *Unbiased standard deviation*

Description

This function computes the unbiased standard deviation of the values in x. If na.rm is TRUE, then missing values are removed before computation proceeds.

Usage

```
sd.unbiased(x, na.rm = FALSE)
```

Arguments

x	a numeric vector or an R object which is coercible to one by as.double(x).
na.rm	logical. If TRUE, then missing values are removed before computation proceeds.

Details

sd is not unbiased. This function computes sd divided by $c_4(n)$, where $c_4(n)$ is obtained by c4.factor(n, estimator="sd"){rQCC}.

Author(s)

Chanseok Park

See Also

sd for the square root of var, but it is biased.

Refer to mad.unbiased{rQCC} for finite-sample unbiased median absolute deviation (MAD) estimator, the most robust alternative.

Examples

```
sd.unbiased(1:2)
```

Shamos   *Shamos estimator*

Description

Calculates the conventional Shamos, unbiased Shamos and unbiased squared Shamos estimators. The conventional Shamos is calculated by shamos which is Fisher-consistent under the normal distribution. Note that it is not unbiased with a sample of finite size. The unbiased Shamos estimator under the normal distribution is calculated by shamos.unbiased with a finite-sample unbiasing factor. The unbiased *squared* Shamos estimator under the normal distribution is calculated by shamos2.unbiased with a finite-sample unbiasing factor.

Usage

```
shamos(x, constant=1.048358, na.rm = FALSE,  IncludeEqual=FALSE)

shamos.unbiased(x, constant=1.048358, na.rm = FALSE,  IncludeEqual=FALSE)

shamos2.unbiased(x, constant=1.048358, na.rm = FALSE, IncludeEqual=FALSE)
```

Arguments

x	a numeric vector of observations.
constant	Correction factor for the Fisher-consistency under the normal distribution
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
IncludeEqual	FALSE (default) calculates median of $|X_i - X_j|$ with $i < j$, while TRUE calculates median of $|X_i - X_j|$ with $i \leq j$.

Details

The Shamos estimator is defined as

$$\text{Shamos} = \texttt{constant} \times \underset{i<j}{\text{median}} \left(|X_i - X_j| \right)$$

where $i, j = 1, 2, \ldots, n$. The default value (constant=1.048358) ensures the Fisher-consistency under the normal distribution. Note that $\texttt{constant} = 1/\{\sqrt{2}\,\Phi^{-1}(3/4)\} \approx 1.048358$.

The unbiased Shamos is defined as

$$\text{Shamos} = \texttt{constant} \times \underset{i<j}{\text{median}}\left(|X_i - X_j|\right)/c_6(n)$$

for $i, j = 1, 2, \ldots, n$, where $c_6(n)$ is the finite-sample unbiasing factor. Note that $c_6(n)$ notation is used in Park et. al (2022), and $c_6(n)$ is calculated using the function c4.factor{rQCC} with estimator="shamos" option.

The unbiased squared Shamos is defined as the squared shamos{rQCC} divided by $w_6(n)$ where $w_6(n)$ is the finite-sample unbiasing factor. Note that $w_6(n)$ notation is used in Park et. al (2022), and $w_6(n)$ is calculated using the function w4.factor{rQCC} with estimator="shamos2" option. Note that the square of the conventional Shamos estimator is Fisher-consistent for the variance (σ^2) under the normal distribution, but it is not unbiased with a sample of finite size.

Value

They return a numeric value.

Author(s)

Chanseok Park and Min Wang

References

Park, C., H. Kim, and M. Wang (2022). Investigation of finite-sample properties of robust location and scale estimators. *Communications in Statistics - Simulation and Computation*, **51**, 2619-2645. doi:10.1080/03610918.2019.1699114

Shamos, M. I. (1976). Geometry and statistics: Problems at the interface. In Traub, J. F., editor, *Algorithms and Complexity: New Directions and Recent Results*, pages 251–280. Academic Press, New York.

Lèvy-Leduc, C., Boistard, H., Moulines, E., Taqqu, M. S., and Reisen, V. A. (2011). Large sample behaviour of some well-known robust estimators under long-range dependence. *Statistics*, **45**, 59–71.

See Also

mad.unbiased{rQCC} for calculating the *unbiased* sample MAD.

mad{stats} for calculating the Fisher-consistent sample MAD.

c4.factor{rQCC} for finite-sample unbiasing factor for the standard deviation (σ) under the normal distribution.

w4.factor{rQCC} for finite-sample unbiasing factor for the squared Shamos estimator of the variance (σ^2) under the normal distribution.

finite.breakdown{rQCC} for calculating the finite-sample breakdown point.

Examples

```
x = c(0:10, 50)

# Fisher-consistent Shamos, but not unbiased with a finite sample.
shamos(x)

# Unbiased Shamos.
shamos.unbiased(x)

# Fisher-consistent squared Shamos, but not unbiased with a finite sample.
shamos(x)^2

# Unbiased squared Shamos.
shamos2.unbiased(x)
```

---

unbiasing.factor   *Finite-sample unbiasing factor*

Description

Finite-sample unbiasing factor for estimating the standard deviation (σ) and the variance (σ^2) under the normal distribution.

Usage

```
c4.factor(n, estimator=c("sd","range", "mad","shamos"))
w4.factor(n, estimator=c("mad2","shamos2"))
```

Arguments

n	sample size ($n \geq 1$).
estimator	a character string specifying the estimator, must be one of "sd" (default), "range", "mad", "shamos" for c4.factor, and one of "mad2" (default), "shamos2" for w4.factor.

Details

The conventional sample standard deviation, range, median absolute deviation (MAD) and Shamos estimators are Fisher-consistent under the normal distribution, but they are not unbiased with a sample of finite size.

Using the sample standard deviation, an unbiased estimator of the standard deviation (σ) is calculated by
```
sd(x)/c4.factor(length(x), estimator="sd")
```

Using the range (maximum minus minimum), an unbiased estimator of σ is calculated by
```
diff(range(x))/c4.factor(length(x), estimator="range")
```

Using the median absolute deviation (mad{stats}), an unbiased estimator of σ is calculated by
mad(x)/c4.factor(length(x), estimator="mad")

Using the Shamos estimator (shamos{rQCC}), an unbiased estimator of σ is calculated by
shamos(x)/c4.factor(length(x), estimator="shamos")

Note that the formula for the unbiasing factor $c_4(n)$ is given by

$$c_4(n) = \sqrt{\frac{2}{n-1}} \cdot \frac{\Gamma(n/2)}{\Gamma((n-1)/2)}.$$

The *squared* MAD and *squared* Shamos are Fisher-consistent for the variance (σ^2) under the normal distribution, but they are not unbiased with a sample of finite size.

An unbiased estimator of the variance (σ^2) is obtained using the finite-sample unbiasing factor (w4.factor).

Using the squared MAD, an unbiased estimator of σ^2 is calculated by
mad(x)^2/w4.factor(length(x), estimator="mad2")

Using the squared Shamos estimator, an unbiased estimator of σ^2 is calculated by
shamos(x)^2/w4.factor(length(x), estimator="shamos2")

The finite-sample unbiasing factors for the median absolute deviation (MAD) and Shamos estimators are obtained for $n = 1, 2, \ldots, 100$ using the extensive Monte Carlo simulation with 1E07 replicates. For the case of $n > 100$, they are obtained using the method of Hayes (2014).

Value

It returns a numeric value.

Author(s)

Chanseok Park

References

Park, C., H. Kim, and M. Wang (2022). Investigation of finite-sample properties of robust location and scale estimators. *Communications in Statistics - Simulation and Computation*, **51**, 2619-2645. doi:10.1080/03610918.2019.1699114

Shamos, M. I. (1976). Geometry and statistics: Problems at the interface. In Traub, J. F., editor, *Algorithms and Complexity: New Directions and Recent Results*, 251–280. Academic Press, New York.

Hayes, K. (2014). Finite-sample bias-correction factors for the median absolute deviation. *Communications in Statistics: Simulation and Computation*, **43**, 2205–2212.

See Also

mad{stats} for the Fisher-consistent median absolute deviation (MAD) estimator of the standard deviation (σ) under the normal distribution.

`mad.unbiased{rQCC}` for finite-sample unbiased median absolute deviation (MAD) estimator of the standard deviation (σ) under the normal distribution.

`shamos{rQCC}` for the Fisher-consistent Shamos estimator of the standard deviation (σ) under the normal distribution.

`shamos.unbiased{rQCC}` for finite-sample unbiased Shamos estimator of the standard deviation (σ) under the normal distribution.

`n.times.eBias.of.mad{rQCC}` for the values of the empirical bias of the median absolute deviation (MAD) estimator under the standard normal distribution.

`n.times.eBias.of.shamos{rQCC}` for the values of the empirical bias of the Shamos estimator under the standard normal distribution.

`mad2.unbiased{rQCC}` for finite-sample unbiased squared MAD estimator of the variance (σ^2) under the normal distribution.

`shamos2.unbiased{rQCC}` for finite-sample unbiased squared Shamos estimator of the variance (σ^2) under the normal distribution.

`n.times.evar.of.mad{rQCC}` for the values of the empirical variance of the median absolute deviation (MAD) estimator under the standard normal distribution.

`n.times.evar.of.shamos{rQCC}` for the values of the empirical variance of the Shamos estimator under the standard normal distribution.

Examples

```
# unbiasing factor for estimating the standard deviation
c4.factor(n=10, estimator="sd")
c4.factor(n=10, estimator="mad")
c4.factor(n=10, estimator="shamos")

# Note: d2 notation is widely used for the bias-correction of the range.
d2 = c4.factor(n=10, estimator="range")
d2

# unbiasing factor for estimating the variance
w4.factor(n=10, "mad2")
w4.factor(n=10, "shamos2")
```

Index