Package: CUSUMdesign (via r-universe)

September 15, 2024

Type Package

Title Compute Decision Interval and Average Run Length for CUSUM Charts

Version 1.1.5

Date 2020-02-22

Author Douglas M. Hawkins, David H. Olwell, and Boxiang Wang

Maintainer Boxiang Wang <boxiang-wang@uiowa.edu>

Description Computation of decision intervals (H) and average run lengths (ARL) for CUSUM charts. Details of the method are seen in Hawkins and Olwell (2012): Cumulative sum charts and charting for quality improvement, Springer Science & Business Media.

License GPL-2

NeedsCompilation yes

Repository https://boxiang-wang.r-universe.dev

RemoteUrl https://github.com/boxiang-wang/cusumdesign

RemoteRef HEAD

RemoteSha a1b4271cfad68ee737d0de88c43d0cb2144b1768

Contents

getARL 2
getH 4

Index **7**

getARL *compute average run length (ARL) for CUSUM charts*

Description

Compute average run lengths for CUSUM charts based on the Markov chain algorithm.

Usage

```
getARL(distr=NULL, K=NULL, H=NULL,
    Mean=NULL, std=NULL, prob=NULL, Var=NULL, mu=NULL, lambda=NULL,
    samp.size=NULL, is.upward=NULL, winsrl=NULL, winsru=NULL)
```

Arguments

distr	Integer valued from 1 to 6: 1 refers to “normal mean", 2 refers to “normal variance", 3 refers to “Poisson", 4 refers to “binomial", 5 refers to “negative binomial", and 6 refers to “inverse Gaussian mean".
K	A reference value, which is given by getH.
H	A given decision interval, which is given by getH.
Mean	Mean value, which has to be provided when distr = 1 (normal mean), 3 (Poisson), and 5 (negative binomial). The value must be positive when distr = 3 or distr = 5.
std	Standard deviation, which has to be provided when distr = 1 (normal mean) and 2 (normal variance). The value must be positive.
prob	Success probability, which has to be provided when distr = 4 (binomial); 0 < prob <= 1.
Var	Variance, which has to be provided when distr = 5 (negative binomial). The value has to be larger than Mean when distr = 5.
mu	A positive value representing the mean of inverse Gaussian distribution. The argument ’mu’ has to be provided when distr = 6 (inverse Gaussian mean).
lambda	A positive value representing the shape parameter for inverse Gaussian distribution. The argument ’lambda’ has to be provided when distr = 6 (inverse Gaussian mean).
samp.size	Sample size, an integer which has to be provided when distr = 2 (normal variance) or distr = 4 (binomial).
is.upward	Logical value, whether to depict a upward or downward CUSUM.
winsrl	Lower Winsorizing constant. Use NULL or -999 if Winsorization is not needed.
winsru	Upper Winsorizing constant. Use NULL or 999 if Winsorization is not needed.

Details

Computes ARL when the reference value and decision interval are given. For each case, the necessary parameters are listed as follows.

Normal mean (distr = 1): `Mean`, `std`, `K`, `H`.
Normal variance (distr = 2): `samp.size`, `std`, `K`, `H`.
Poisson (distr = 3): `Mean`, `K`, `H`.
Binomial (dist = 4): `samp.size`, `prob`, `K`, `H`.
Negative binomial (distr = 5): `Mean`, `Var`, `K`, `H`.
Inverse Gaussian mean (distr = 6): `mu`, `lambda`, `K`, `H`.

Value

A list including three variables:

`ARL_Z`	The computed zero-start average run length for CUSUM.
`ARL_F`	The computed fast-initial-response (FIR) average run length for CUSUM.
`ARL_S`	The computed steady-state average run length for CUSUM.

Author(s)

Douglas M. Hawkins, David H. Olwell, and Boxiang Wang
Maintainer: Boxiang Wang <boxiang-wang@uiowa.edu>

References

Hawkins, D. M. and Olwell, D. H. (1998) “Cumulative Sum Charts and Charting for Quality Improvement (Information Science and Statistics)", Springer, New York.

See Also

`getH`

Examples

```
# normal mean
getARL(distr=1, K=11, H=5, Mean=10, std=2)

# normal variance
getARL(distr=2, K=3, H=1, std=2, samp.size=5, is.upward=TRUE)

# Poission
getARL(distr=3, K=3, H=1, std=2, Mean=5, is.upward=TRUE)

# Binomial
getARL(distr=4, K=0.8, H=1, prob=0.2, samp.size=100, is.upward=TRUE)

# Negative binomial
getARL(distr=5, K=3, H=6, Mean=2, Var=5, is.upward=TRUE)
```

```
# Inverse Gaussian mean
getARL(distr=6, K=2, H=4, mu=3, lambda=0.5, is.upward=TRUE)
```

---

getH *compute decision interval (H) for CUSUM charts*

Description

Compute decision intervals for CUSUM charts.

Usage

```
getH(distr=NULL, ARL=NULL, ICmean=NULL, ICsd=NULL,
    OOCmean=NULL, OOCsd=NULL, ICprob=NULL, OOCprob=NULL,
    ICvar=NULL, IClambda=NULL, samp.size=NULL,
    ref=NULL, winsrl=NULL, winsru=NULL,
    type=c("fast initial response", "zero start", "steady state"))
```

Arguments

distr	Integer valued from 1 to 6: 1 refers to “normal mean", 2 refers to “normal variance", 3 refers to “Poisson", 4 refers to “binomial", 5 refers to “negative binomial", 6 refers to “inverse Gaussian mean".
ARL	An integer for in control average run length.
ICmean	In-control mean, which has to be provided when distr = 1 (normal mean), 3 (Poisson), 5 (negative binomial), and 6 (inverse Gaussian mean). The value has to be positive when distr = 3, distr = 5, or distr = 6.
ICsd	In-control standard deviation, which has to be provided when distr = 1 (normal mean) and 2 (normal variance). The value has to be positive.
OOCmean	Out-of-control mean, which has to be provided when distr = 1 (normal mean), 3 (Poisson), 5 (negative binomial), and 6 (Inverse Gaussian mean). When distr = 3, 5, or 6, the value has to be positive.
OOCsd	Out-of-control standard deviation, which has to be provided when distr = 2 (normal variance). The value has to be positive.
ICprob	In-control success probability, which has to be provided when distr = 4 (binomial); 0 < prob <= 1.
OOCprob	Out-of-control success probability, which has to be provided when distr = 4 (binomial); 0 < prob <= 1.
ICvar	In-control variance, which has to be provided when distr = 5 (negative binomial). The value has to be larger than the in-control mean ’ICmean’.
IClambda	In-control shape parameter for inverse Gaussian distribution. The argument ’IClambda’ has to be provided when distr = 6 (inverse Gaussian mean).
samp.size	Sample size, an integer which has to be provided when distr = 2 (normal variance) or distr = 4 (binomial).

ref	Optional reference value.
winsrl	Lower Winsorizing constant. Use NULL or -999 if Winsorization is not needed.
winsru	Upper Winsorizing constant. Use NULL or 999 if Winsorization is not needed.
type	A string for CUSUM type: "F" for fast-initial-response CUSUM, "Z" for zero-start CUSUM, and "S" for steady-state CUSUM. Default is "F".

Details

Computes the decision interval H when the reference value and the average run length are given. For each case, the necessary parameters are listed as follows.

Normal mean (distr = 1): ICmean, ICsd, OOCmean.
Normal variance (distr = 2): samp.size, ICsd, OOCsd
Poisson (distr = 3): ICmean, OOCmean.
Binomial (dist = 4): samp.size, ICprob, OOCprob.
Negative binomial (distr = 5): ICmean, Icvar, OOCmean.
Inverse Gaussian mean (distr = 6): ICmean, IClambda, OOCmean.

Value

A list including three variables:

DI	Decision interval.
IC_ARL	In-control average run length.
OOCARL_Z	Out-of-control average run length for the zero-start CUSUM.
OOCARL_F	Out-of-control average run length for the fast-initial-response (FIR) CUSUM.
OOCARL_S	Out-of-control average run length for the steady-state CUSUM.

Author(s)

Douglas M. Hawkins, David H. Olwell, and Boxiang Wang
Maintainer: Boxiang Wang <boxiang-wang@uiowa.edu>

References

Hawkins, D. M. and Olwell, D. H. (1998) “Cumulative Sum Charts and Charting for Quality Improvement (Information Science and Statistics)", Springer, New York.

See Also

getARL

Examples

```
# normal mean
getH(distr=1, ICmean=10, ICsd=2, OOCmean=15, ARL=1000, type="F")

# normal variance
getH(distr=2, ICsd=2, OOCsd=4, samp.size=5, ARL=1000, type="F")

# Poission
getH(distr=3, ICmean=2, OOCmean=3, ARL=100, type="F")

# Binomial
getH(distr=4, ICprob=0.2, OOCprob=0.6, samp.size=100, ARL=1000, type="F")

# Negative binomial
getH(distr=5, ICmean=1, ICvar=3, OOCmean=2, ARL=100, type="F")

# Inverse Gaussian mean
getH(distr=6, ICmean=1, IClambda=0.5, OOCmean=2, ARL=1000, type="F")
```

Index

∗ **ARL**
getARL, 2
∗ **CUSUM**
getARL, 2
getH, 4
∗ **Decision interval**
getH, 4
∗ **Quality control**
getARL, 2
getH, 4

getARL, 2, *5*
getH, *2, 3*, 4