Package ‘LGCU’

July 21, 2025

Title Implementation of Learning Gamma CUSUM (Cumulative Sum) Control Charts

Version 0.1.5

Description Implements Cumulative Sum (CUSUM) control charts specifically designed for monitoring processes following a Gamma distribution. Provides functions to estimate distribution parameters, simulate control limits, and apply cautious learning schemes for adaptive thresholding. It supports upward and downward monitoring with guaranteed performance evaluated via Monte Carlo simulations. It is useful for quality control applications in industries where data follows a Gamma distribution. Methods are based on Madrid-Alvarez et al. (2024) <doi:10.1002/qre.3464> and Madrid-Alvarez et al. (2024) <doi:10.1080/08982112.2024.2440368>.

Depends R (>= 4.0.0)

Imports MASS, Rcpp, tictoc

LinkingTo Rcpp

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

URL `https://github.com/ingharold-madrid/LGCU`

BugReports `https://github.com/ingharold-madrid/LGCU/issues`

NeedsCompilation yes

Author Harold Manuel Madrid-Alvarez [aut, cre],
Victor Gustavo Tercero-Gomez [aut],
Juan Carlos Garcia-Diaz [aut]

Maintainer Harold Manuel Madrid-Alvarez <`harold.madrid@unisimon.edu.co`>

Repository CRAN

Date/Publication 2025-03-17 00:20:06 UTC

Contents

`ARL_Clminus` *ARL Estimation in CUSUM Control Charts with Gamma Distribution and Cautious Learning for downward detection*

Description

This function calculates the **Average Run Length (ARL)** of a CUSUM control chart based on the Gamma distribution, incorporating a cautious learning scheme for the dynamic update of parameters.

The function allows the evaluation of the CUSUM chart's performance under different parameterization scenarios, ensuring efficient detection of process changes.

Based on the methodology presented in the work of Madrid-Alvarez, García-Díaz, and Tercero-Gómez (2024), this implementation uses Monte Carlo simulations optimized in C++ for efficient execution and progressive adjustment of the control chart parameters.

The values for `H_minus`, `H_delta`, `K_l`, `delay`, and `tau` can be referenced in the tables from the article:

Madrid-Alvarez, H. M., García-Díaz, J. C., & Tercero-Gómez, V. G. (2024). *A CUSUM control chart for the Gamma distribution with cautious parameter learning.* Quality Engineering, 1-23.

Usage Scenarios::

Scenario 1: Known `alpha` and estimated `beta`

- The `alpha` parameter is assumed to be fixed and known in advance.
- `beta` is estimated from a dataset or provided by the user.
- The user must specify `alpha` and an initial estimate of `beta` (`beta0_est`).

Scenario 2: Both `alpha` and `beta` are estimated

- Both `alpha` and `beta` are estimated from an external dataset.

- The user must calculate alpha0_est and beta0_est before calling the function.
- beta0_est is dynamically updated during the simulation when a predefined condition is met.

Features::

- Implements Monte Carlo simulations for ARL estimation.
- Allows dynamic updating of beta0_est to improve model adaptation.
- Uses C++ optimization for efficient and precise execution.
- Compatible with scenarios where alpha is either known or estimated.
- Recommended values for H_minus, H_delta, K_l, delay, and tau can be found in the reference article.

This function is ideal for quality control studies where reliable detection of process changes modeled with Gamma distributions is required.

Usage

```
ARL_Clminus(
  alpha,
  beta,
  alpha0_est,
  beta0_est,
  known_alpha,
  beta_ratio,
  H_delta,
  H_minus,
  n_I,
  replicates,
  K_l,
  delay,
  tau
)
```

Arguments

alpha	Shape parameter of the Gamma distribution.
beta	Scale parameter of the Gamma distribution.
alpha0_est	Initial estimate of the shape parameter alpha. If known_alpha is TRUE, this value will be equal to alpha.
beta0_est	Initial estimate of the scale parameter beta. This value is updated dynamically during the simulation.
known_alpha	TRUE if alpha0_est is fixed, FALSE if it must be estimated.
beta_ratio	Ratio between beta and its posterior estimate.
H_delta	Increment of the lower control limit in the CUSUM chart.
H_minus	Initial control limit of the CUSUM chart for downward detection.
n_I	Sample size in Phase I.
replicates	Number of Monte Carlo simulations.

K_l	Secondary control threshold for parameter updating.
delay	Number of observations before updating beta0_est.
tau	Time point at which beta changes.

Value

A numeric value corresponding to the **ARL** estimate for the downward CUSUM control chart with cautious learning.

Examples

```
# Option 1: Provide parameters directly
ARL_Clminus(
   alpha = 1,
   beta = 1,
   alpha0_est = 1.067,  # alpha = known_alpha
   beta0_est = 0.2760,   # Estimated Beta
   known_alpha = TRUE,
   beta_ratio = 1/2,
   H_delta = 0.6946,
   H_minus = -4.8272,
   n_I = 500,
   replicates = 1000,
   K_l = 0.5,
   delay = 25,
   tau = 1
)

# Option 2: Use generated data
set.seed(123)
datos_faseI <- rgamma(n = 500, shape = 1, scale = 1)
alpha0_est <- mean(datos_faseI)^2 / var(datos_faseI)  # Alpha estimation
beta0_est <- mean(datos_faseI) / alpha0_est  # Beta estimation

ARL_Clminus(
   alpha = 1,
   beta = 1,
   alpha0_est = 1.067,  # alpha = known_alpha
   beta0_est = 0.2760,   # Estimated Beta
   known_alpha = FALSE,
   beta_ratio = 1/2,
   H_delta = 0.6946,
   H_minus = -4.8272,
   n_I = 500,
   replicates = 1000,
   K_l = 0.5,
   delay = 25,
   tau = 1
)
```

ARL_Clplus *ARL Estimation in CUSUM Control Charts with Gamma Distribution and Cautious Learning for upward detection*

Description

This function calculates the **Average Run Length (ARL)** of a CUSUM control chart based on the Gamma distribution, incorporating a cautious learning scheme for the progressive update of parameters and optimization of performance in upward detection.

The function allows for the evaluation of the CUSUM chart's behavior under different parameterization scenarios, ensuring efficient detection of process changes.

Following the methodology presented in the work of Madrid-Alvarez, García-Díaz, and Tercero-Gómez (2024), this implementation utilizes Monte Carlo simulations in C++ for efficient execution, ensuring a dynamic adjustment of parameters based on the evolution of the process.

The values of `H_plus`, `H_delta`, `K_l`, `delay`, and `tau` can be referenced in the tables from the article:

Madrid-Alvarez, H. M., García-Díaz, J. C., & Tercero-Gómez, V. G. (2024). *A CUSUM control chart for the Gamma distribution with cautious parameter learning.* Quality Engineering, 1-23.

Usage Scenarios::

Scenario 1: Known `alpha` and estimated `beta`

- The `alpha` parameter is assumed to be fixed and known in advance.
- `beta` is estimated from a dataset or defined by the user.
- The user must provide `alpha` and an initial estimate of `beta` (`beta0_est`).

Scenario 2: Both `alpha` and `beta` are estimated

- Both `alpha` and `beta` are estimated from a dataset or external data source.
- The user must calculate `alpha0_est` and `beta0_est` before running the function.
- `beta0_est` is dynamically updated during the simulation when a predefined condition is met.

Features::

- Implements Monte Carlo simulations optimized in C++ for ARL estimation.
- Allows dynamic updating of `beta0_est` to improve the model's adaptability.
- Compatible with scenarios where `alpha` is known or estimated.
- Ensures stable and reliable performance in detecting changes in processes modeled with Gamma distributions.
- Recommended values for `H_plus`, `H_delta`, `K_l`, `delay`, and `tau` can be found in the reference article.

Usage

```
ARL_Clplus(
  alpha,
  beta,
```

```
  alpha0_est,
  beta0_est,
  known_alpha,
  beta_ratio,
  H_delta,
  H_plus,
  n_I,
  replicates,
  K_l,
  delay,
  tau
)
```

Arguments

`alpha`	Shape parameter of the Gamma distribution.
`beta`	Scale parameter of the Gamma distribution.
`alpha0_est`	Initial estimate of the shape parameter `alpha`. If `known_alpha` is `TRUE`, this value will be equal to `alpha`.
`beta0_est`	Initial estimate of the scale parameter `beta`. This value is updated dynamically during the simulation.
`known_alpha`	`TRUE` if `alpha0_est` is fixed, `FALSE` if it must be estimated.
`beta_ratio`	Ratio between `beta` and its posterior estimate.
`H_delta`	Increment of the upper control limit in the CUSUM chart.
`H_plus`	Initial control limit of the CUSUM chart.
`n_I`	Sample size in Phase I.
`replicates`	Number of Monte Carlo simulations.
`K_l`	Secondary control threshold for parameter updating.
`delay`	Number of observations before updating `beta0_est`.
`tau`	Time point at which `beta` changes. A value of 1 is recommended for IC states.

Value

A numeric value corresponding to the **ARL** estimate for the upward CUSUM control chart with cautious learning.

Examples

```
# Option 1: Provide parameters directly
ARL_Clplus(
  alpha = 1,
  beta = 1,
  alpha0_est = 1,  # alpha = known_alpha
  beta0_est = 1.1,   # Estimated Beta
  known_alpha = TRUE,
  beta_ratio = 2,
```

```
  H_delta = 4.2433,
  H_plus = 8.7434,
  n_I = 200,
  replicates = 100,
  K_l = 2,
  delay = 25,
  tau = 1
)

# Option 2: Use generated data
set.seed(123)
datos_faseI <- rgamma(n = 200, shape = 1, scale = 1)
alpha0_est <- mean(datos_faseI)^2 / var(datos_faseI)  # Alpha estimation
beta0_est <- mean(datos_faseI) / alpha0_est  # Beta estimation

ARL_Clplus(
  alpha = 1,
  beta = 1,
  alpha0_est = alpha0_est,
  beta0_est = beta0_est,
  known_alpha = FALSE,
  beta_ratio = 2,
  H_delta = 4.2433,
  H_plus = 8.7434,
  n_I = 200,
  replicates = 1000,
  K_l = 2,
  delay = 25,
  tau = 1
)
```

`getDeltaHL_down` *Estimation of the* `H_delta` *parameter with learning for downward detection in CUSUM Gamma charts*

Description

This function calculates the optimal value of `H_delta` using a dynamic learning scheme based on the `ARL_Clplus` function, iteratively adjusting `H_delta` to achieve an **expected ARL** with greater accuracy and adaptability.

Based on the methodology proposed by Madrid-Alvarez, Garcia-Diaz, and Tercero-Gomez (2024), this function allows adjusting `H_delta` in different sample size scenarios, ensuring that the control chart progressively adapts to changes in the Gamma distribution.

Features::

- Implements Monte Carlo simulations to estimate `H_delta`.
- Relies on parameter estimates obtained in Phase I.
- Iteratively adjusts `H_delta` until the specified ARL is reached.
- Incorporates a cautious learning mechanism to improve adjustment accuracy.

- Displays total execution time using `tictoc`.

Recommendations:

- This function is useful for estimating `H_delta` values when the sample size differs from those reported in the reference article:
  Madrid-Alvarez, H. M., Garcia-Diaz, J. C., & Tercero-Gomez, V. G. (2024). *A CUSUM control chart for the Gamma distribution with cautious parameter learning.* Quality Engineering, 1-23.
- **The adjustment process is iterative and computationally demanding**, as execution time depends on the number of iterations (`N_init` + `N_final`) and the sample size (`n_I`).
- It is recommended to define an appropriate convergence criterion to optimize execution time without compromising accuracy in the estimation of `H_delta`.
- For selecting values of `a`, `b`, `k_l`, `delay`, `tau`, and `H_minus`, refer to the reference article, which presents specific strategies for their calibration in different scenarios.

Usage

```
getDeltaHL_down(
  n_I,
  alpha,
  beta,
  beta_ratio,
  H_minus,
  a,
  b,
  ARL_esp,
  replicates,
  N_init,
  N_final,
  known_alpha,
  K_l,
  delay,
  tau
)
```

Arguments

`n_I`	Sample size in Phase I.
`alpha`	Shape parameter of the Gamma distribution.
`beta`	Scale parameter of the Gamma distribution.
`beta_ratio`	Ratio between `beta` and its posterior estimate.
`H_minus`	Lower limit of the CUSUM chart.
`a`	Tolerance level for the expected ARL ($0 <= a < 1$).
`b`	Tolerance level for the expected ARL ($0 < b < 1$).
`ARL_esp`	Desired expected ARL value.
`replicates`	Number of replications in the Monte Carlo simulation.

N_init	Initial iterations for adjustment.
N_final	Final iterations for averaging H_delta.
known_alpha	TRUE if alpha is fixed, FALSE if it must be estimated.
K_l	Secondary control threshold for parameter update.
delay	Number of observations before updating beta0_est.
tau	Time point where beta changes.

Value

A numeric value corresponding to the optimal H_delta estimated with learning for the downward CUSUM control chart.

Examples

```
getDeltaHL_down(n_I = 200, alpha = 1, beta = 1, beta_ratio = 1/1.5,
             H_minus = -6.2913, a = 0.1, b = 0.05, ARL_esp = 370,
             replicates = 10, N_init = 100, N_final = 1000,
             known_alpha = TRUE, K_l = 0.7, delay = 25, tau = 1)
```

---

getDeltaHL_up — *Estimation of the* H_delta *parameter with learning for upward detection in Gamma CUSUM control charts*

Description

This function calculates the optimal value of H_delta using a dynamic learning scheme based on the ARL_Clplus function, iteratively adjusting H_delta to achieve an **expected ARL** with higher accuracy and adaptability.

Based on the methodology proposed by Madrid-Alvarez, García-Díaz, and Tercero-Gómez (2024), this function allows adjusting H_delta in different sample size scenarios, ensuring that the control chart progressively adapts to changes in the Gamma distribution.

Features::

- Implements Monte Carlo simulations to estimate H_delta.
- Relies on parameter estimates obtained in Phase I.
- Iteratively adjusts H_delta until the specified ARL is reached.
- Incorporates a cautious learning mechanism to improve adjustment precision.
- Displays the total execution time using tictoc.

Recommendations:

- This function is useful for estimating H_delta values when the sample size differs from the values reported in the reference article:
  Madrid-Alvarez, H. M., García-Díaz, J. C., & Tercero-Gómez, V. G. (2024). *A CUSUM control chart for the Gamma distribution with cautious parameter learning.* Quality Engineering, 1-23.

- **The adjustment process is iterative and computationally intensive**, as execution time depends on the number of iterations (N_init + N_final) and the sample size (n_I).
- It is recommended to define a proper convergence criterion to optimize execution time without compromising the accuracy of H_delta estimation.
- For selecting values of a, b, k_l, delay, tau, and H_plus, consulting the reference article is recommended, as it provides specific strategies for their calibration in different scenarios.

Usage

```
getDeltaHL_up(
  n_I,
  alpha,
  beta,
  beta_ratio,
  H_plus,
  a,
  b,
  ARL_esp,
  replicates,
  N_init,
  N_final,
  known_alpha,
  K_l,
  delay,
  tau
)
```

Arguments

n_I	Sample size in Phase I.
alpha	Shape parameter of the Gamma distribution.
beta	Scale parameter of the Gamma distribution.
beta_ratio	Ratio between beta and its posterior estimate.
H_plus	Initial limit of the CUSUM chart.
a	Tolerance level for the expected ARL. (0 <= a < 1).
b	Tolerance level for the expected ARL. (0 < b < 1)
ARL_esp	Desired expected ARL value.
replicates	Number of replications in the Monte Carlo simulation.
N_init	Number of initial iterations for adjustment.
N_final	Number of final iterations for averaging H_delta.
known_alpha	TRUE if alpha is fixed, FALSE if it should be estimated.
K_l	Secondary control threshold for parameter updating.
delay	Number of observations before updating beta0_est.
tau	Point in time where beta changes.

Value

A numeric value corresponding to the optimal H_delta estimated with learning for the upward CUSUM control chart.

Examples

```
getDeltaHL_up(
           n_I = 200, alpha = 1, beta = 1, beta_ratio = 2,
            H_plus = 6.8313, a = 0.1, b = 0.05, ARL_esp = 370,
             replicates = 100, N_init = 100, N_final = 500,
             known_alpha = TRUE, K_l = 2, delay = 25, tau = 1
            )
```

---

getDeltaH_down — *Estimation of the Optimal* H_delta *Value to Guarantee Performance in the Downward CUSUM Control Chart*

Description

This function calculates the optimal value of H_delta that guarantees a specific performance in the Gamma CUSUM control chart for downward detection. It employs a Monte Carlo simulation approach and an iterative adjustment process to determine the appropriate value.

Following the methodology presented by Madrid-Alvarez, García-Díaz, and Tercero-Gómez (2024), this function allows adjusting H_delta for different sample size configurations, ensuring that the control chart maintains the desired performance in terms of expected ARL.

Features::

- Implements Monte Carlo simulations to estimate H_delta.
- Based on parameter estimates obtained in Phase I.
- Iteratively adjusts H_delta until the specified ARL is achieved.
- Displays the total execution time using tictoc.

Recommendations:

- This function is useful for estimating H_delta values in scenarios where the sample size differs from the values reported in the reference paper:
  Madrid-Alvarez, H. M., García-Díaz, J. C., & Tercero-Gómez, V. G. (2024). *A CUSUM control chart for gamma distribution with guaranteed performance.* Quality and Reliability Engineering International, 40(3), 1279-1301.
- **The adjustment process is iterative and computationally demanding**, as its execution time depends on the number of iterations (N_init + N_final) and the sample size (n_I).
- It is recommended to establish an appropriate convergence criterion to optimize execution time without compromising the accuracy of H_delta estimation.
- For selecting values of H_minus, a, and b, it is advisable to consult the reference paper, which provides specific calibration strategies and recommendations.

Usage

```
getDeltaH_down(
  n_I,
  alpha,
  beta,
  beta_ratio,
  H_minus,
  a,
  b,
  ARL_esp,
  m,
  N_init,
  N_final,
  known_alpha
)
```

Arguments

`n_I`	Sample size in Phase I.
`alpha`	Shape parameter of the Gamma distribution.
`beta`	Scale parameter of the Gamma distribution.
`beta_ratio`	Ratio between beta and its estimate.
`H_minus`	Initial lower limit of the CUSUM chart.
`a`	Tolerance level for the expected ARL (0 <= a < 1).
`b`	Tolerance level for the expected ARL (0 < b < 1).
`ARL_esp`	Desired expected ARL value.
`m`	Number of states in the Markov matrix.
`N_init`	Number of initial iterations.
`N_final`	Number of final iterations.
`known_alpha`	Indicates whether `alpha` is known (`TRUE`) or should be estimated (`FALSE`).

Value

A numerical value corresponding to the optimal `H_delta` for the downward CUSUM control chart, ensuring the expected performance.

Examples

```
getDeltaH_down(n_I = 100, alpha = 1, beta = 1, beta_ratio = 1/2,
               H_minus = -4.1497, a = 0.1, b = 0.05, ARL_esp = 370,
               m = 100, N_init = 10, N_final = 1000, known_alpha = TRUE)
```

getDeltaH_up — *Estimation of the optimal* H_delta *value to guarantee performance in the upward CUSUM control chart*

Description

This function calculates the optimal H_delta value that ensures specific performance in the Gamma CUSUM control chart for upward detection. It relies on Monte Carlo simulations and an iterative adjustment process to determine the appropriate value.

Following the methodology proposed by Madrid-Alvarez, Garcia-Diaz, and Tercero-Gomez (2024), this function allows adjusting H_delta for different sample size scenarios, ensuring that the control chart maintains the expected performance in terms of ARL.

Features::

- Implements Monte Carlo simulations to estimate H_delta.
- Based on parameter estimates obtained in Phase I.
- Iteratively adjusts H_delta until the specified ARL is reached.
- Displays total execution time using tictoc.

Recommendations:

- This function is useful for estimating H_delta values in scenarios where the sample size differs from the values reported in the reference article:
  Madrid-Alvarez, H. M., Garcia-Diaz, J. C., & Tercero-Gomez, V. G. (2024). *A CUSUM control chart for gamma distribution with guaranteed performance.* Quality and Reliability Engineering International, 40(3), 1279-1301.
- **The adjustment process is iterative and computationally demanding**, as its execution time depends on the number of iterations (N_init + N_final) and the sample size (n_I).
- It is recommended to establish an appropriate convergence criterion to optimize execution time without compromising the accuracy of H_delta estimation.
- For selecting values of H_plus, a, and b, refer to the reference article, which presents specific strategies and recommendations for calibration.

Usage

```
getDeltaH_up(
  n_I,
  alpha,
  beta,
  beta_ratio,
  H_plus,
  a,
  b,
  ARL_esp,
  m,
  N_init,
```

```
  N_final,
  known_alpha
)
```

Arguments

n_I	Sample size in Phase I.
alpha	Shape parameter of the Gamma distribution.
beta	Scale parameter of the Gamma distribution.
beta_ratio	Ratio between beta and its estimate.
H_plus	Initial upper limit of the CUSUM chart.
a	Tolerance level for the expected ARL (0 <= a < 1).
b	Tolerance level for the expected ARL (0 < b < 1).
ARL_esp	Desired expected ARL value.
m	Number of states in the Markov matrix.
N_init	Number of initial iterations.
N_final	Number of final iterations.
known_alpha	Indicates whether alpha is known (TRUE) or needs to be estimated (FALSE).

Value

A numeric value corresponding to the optimal H_delta for the upward CUSUM control chart, ensuring the expected performance.

Examples

```
getDeltaH_up(n_I = 100, alpha = 1, beta = 1, beta_ratio = 2, H_plus = 6.8313,
             a = 0.1, b = 0.05, ARL_esp = 370, m = 100,
             N_init = 10, N_final = 1000, known_alpha = TRUE)
```

GICARL_CUSUM_down — *ARL Calculation for a Downward CUSUM Control Chart with a Gamma Distribution*

Description

This function estimates the average run length (**ARL**) of a downward CUSUM control chart applied to a Gamma distribution with guaranteed performance (GIC), considering both known and estimated parameters.

This approach follows the methodology described in the work of Madrid-Alvarez, García-díaz, and Tercero-Gómez(2024), which provides a detailed analysis of the performance of CUSUM control charts for Gamma distributions with guaranteed efficiency. Specifically, the method implemented in this function enables the precise evaluation of ARL under different parameter settings, ensuring appropriate calibration and monitoring of controlled processes.

Recommendations:

For further consultation and to review values of H_delta and H_minus, it is recommended to refer to the following article: Madrid-Alvarez, H. M., García-Díaz, J. C., & Tercero-Gómez, V. G. (2024). **A CUSUM control chart for gamma distribution with guaranteed performance**. Quality and Reliability Engineering.

Considerations::

- The control chart is calibrated with beta = 1.
- When alpha and beta are known, it is recommended to use the same values for alpha_est = alpha and beta_est = 1.
- Higher values of m increase the accuracy of the results (m = {10, 100}).

Usage

```
GICARL_CUSUM_down(
  alpha,
  beta,
  alpha_est,
  beta_est,
  beta_ratio,
  H_minus,
  H_delta,
  m
)
```

Arguments

alpha	Shape parameter of the Gamma distribution.
beta	Scale parameter of the Gamma distribution.
alpha_est	Estimated shape parameter.
beta_est	Estimated scale parameter.
beta_ratio	Ratio between beta and its estimation.
H_minus	Lower control limit of the downward CUSUM chart.
H_delta	Increment of the GIC threshold.
m	Number of divisions for the probability matrix.

Value

A numeric value representing the average run length (**ARL**) of the downward CUSUM control chart.

Examples

```
# Example with known parameters
GICARL_CUSUM_down(alpha = 1, beta = 1, alpha_est = 1, beta_est = 1,
                  beta_ratio = 1/2.5, H_minus = -2.792, H_delta = 0, m = 100)
```

```
# Example with estimated parameters
GICARL_CUSUM_down(alpha = 1, beta = 1, alpha_est = 1, beta_est = 1.1,
                  beta_ratio = 1/2, H_minus = -4.1497, H_delta = 1.5167,
                  m = 100)
```

GICARL_CUSUM_up — *ARL Calculation for an Upward CUSUM Control Chart with a Gamma Distribution*

Description

This function estimates the average run length (**ARL**) of an upward CUSUM control chart applied to a Gamma distribution with guaranteed performance (GIC), considering both known and estimated parameters.

This approach follows the methodology described in the work of Madrid-Alvarez, García-Díaz, and Tercero-Gómez (2024), which provides a detailed analysis of the performance of CUSUM control charts for Gamma distributions with guaranteed efficiency. Specifically, the method implemented in this function enables the precise evaluation of ARL under different parameter settings, ensuring appropriate calibration and monitoring of controlled processes.

Recommendations:

For further consultation and to review values of `H_delta` and `H_plus`, it is recommended to refer to the following article: Madrid-Alvarez, H. M., García-Díaz, J. C., & Tercero-Gómez, V. G. (2024). **A CUSUM control chart for gamma distribution with guaranteed performance**. Quality and Reliability Engineering.

Key Considerations::

- The control chart is calibrated with `beta = 1`.
- When `alpha` and `beta` are known, it is recommended to use the same values for `alpha_est = alpha` and `beta_est = 1`.
- Higher values of `m` increase the accuracy of the results (`m = {10, 100}`).

Usage

```
GICARL_CUSUM_up(
  alpha,
  beta,
  alpha_est,
  beta_est,
  beta_ratio,
  H_plus,
  H_delta,
  m
)
```

Arguments

`alpha`	Shape parameter of the Gamma distribution.
`beta`	Scale parameter of the Gamma distribution.
`alpha_est`	Estimated shape parameter.
`beta_est`	Estimated scale parameter.
`beta_ratio`	Ratio between `beta` and its estimation.
`H_plus`	Upper control limit of the upward CUSUM chart.
`H_delta`	Increment of the GIC threshold.
`m`	Number of divisions for the probability matrix.

Value

A numeric value representing the average run length (**ARL**) of the upward CUSUM control chart.

Examples

```
# Example with known parameters
GICARL_CUSUM_up(alpha = 0.9, beta = 2.136, alpha_est = 0.9, beta_est = 1,
                beta_ratio = 2.67, H_plus = 25.1592, H_delta = 0, m = 100)

# Example with estimated parameters
GICARL_CUSUM_up(alpha = 1, beta = 1, alpha_est = 1.2, beta_est = 0.8,
                beta_ratio = 2, H_plus = 6.5081, H_delta = 2.9693, m = 100)
```

`plot_GICCdown_chart` *Downward CUSUM Control Chart for Gamma Distribution with Guaranteed Performance*

Description

This function generates a downward CUSUM control chart for a Gamma distribution, displaying the evolution of the CUSUM statistic, control limits, and a summary of the parameters.

Based on the approach presented by Madrid-Alvarez, García-Díaz, and Tercero-Gómez (2024), this implementation allows for the evaluation and visualization of monitored processes using a CUSUM chart adapted to Gamma distributions with guaranteed performance.

In particular, the function incorporates a Monte Carlo model to simulate the behavior of the control chart, allowing for the estimation of the Gamma distribution in Phase I or the use of predefined values. Additionally, it provides a clear graphical representation of the evolution of the CUSUM statistic, ensuring appropriate calibration and process control.

Recommendations:

To reference specific values of `H_delta` and `H_minus`, it is recommended to consult the following article: Madrid-Alvarez, H. M., García-Díaz, J. C., & Tercero-Gómez, V. G. (2024). **A CUSUM control chart for gamma distribution with guaranteed performance**. Quality and Reliability Engineering International, 40(3), 1279-1301.

Features::

- Based on a Monte Carlo model.
- The Gamma distribution is either estimated in Phase I or predefined values are used.
- Accumulated values of the CUSUM statistic with guaranteed performance are plotted.
- Control limits and a summary table are included.

Usage

```
plot_GICCdown_chart(
  alpha,
  beta,
  beta_ratio,
  H_delta,
  H_minus,
  n_I,
  n_II,
  faseI = NULL,
  faseII = NULL,
  known_alpha
)
```

Arguments

alpha	Shape parameter of the Gamma distribution.
beta	Scale parameter of the Gamma distribution.
beta_ratio	Ratio between beta and its estimation.
H_delta	Increment of the lower control limit (GIC).
H_minus	Initial lower control limit of the CUSUM chart.
n_I	Sample size in Phase I (if `faseI` is not provided).
n_II	Sample size in Phase II (if `faseII` is not provided).
faseI	Data sample from Phase I (numeric vector). If `NULL`, it is generated using `rgamma()`.
faseII	Data sample from Phase II (numeric vector). If `NULL`, it is generated using `rgamma()`.
known_alpha	If `TRUE`, a known `alpha` is used; if `FALSE`, it is estimated.

Value

A plot showing the evolution of the downward CUSUM statistic, including:

- The accumulated values of the CUSUM statistic.
- Control limits with guaranteed performance.
- A summary of the parameters used in the control chart.

Examples

```
# Option 1: Automatically generate data with defined sample sizes
plot_GICCdown_chart(
                    alpha = 3, beta = 1, beta_ratio = 1/2, H_delta = 0.9596,
                    H_minus = -4.6901,n_I = 100, n_II = 200, faseI = NULL,
                    faseII = NULL, known_alpha = FALSE
                    )

# Option 2: Use custom data
phaseI_data <- rgamma(n = 100, shape = 1, scale = 1)
phaseII_data <- rgamma(n = 200, shape = 1, scale = 1)
plot_GICCdown_chart(
                    alpha = 1, beta = 1, beta_ratio = 1/2, H_delta = 2.9693,
                    H_minus = -6.5081, n_I = 100, n_II = 200,
                    faseI = phaseI_data, faseII = phaseII_data,
                    known_alpha = TRUE
                    )
```

---

plot_GICCLdown_Chart *Downward CUSUM Control Chart with Cautious Learning and Guaranteed Performance*

Description

This function generates a downward CUSUM control chart for a Gamma distribution, incorporating a cautious parameter updating mechanism based on guaranteed performance.

It enables dynamic process monitoring, ensuring progressive adaptation to changes in the distribution.

This approach follows the methodology presented in the work of Madrid-Alvarez, García-Díaz, and Tercero-Gómez (2024), where a cautious learning scheme for parameter updating in CUSUM control charts applied to Gamma distributions is proposed.

The implementation captures changes in the distribution and adjusts the control limits to enhance the detection of process variations.

Features::

- If the user does not provide Phase I and Phase II data, the function generates them automatically.
- If `known_alpha = TRUE`, `alpha` is fixed and not estimated.
- If `known_alpha = FALSE`, `alpha` is estimated from Phase I data.
- Dynamic control limits and a summary of parameters are included.
- Integrates a cautious learning scheme using the parameters `k_l`, `delay`, and `tau`.

Recommendations:

The parameters `k_l`, `delay`, and `tau` are part of the cautious learning mechanism of the CUSUM chart. These values enable the dynamic updating of `beta0_est` and `H_minus`, ensuring that the control chart progressively adapts to process changes, improving sensitivity in detecting deviations.

For proper implementation, it is recommended to reference the values proposed in:

Madrid-Alvarez, H. M., García-Díaz, J. C., & Tercero-Gómez, V. G. (2024). *A CUSUM control chart for the Gamma distribution with cautious parameter learning.* Quality Engineering, 1-23.

While these parameters have been tested and validated in the referenced article, users can adjust them based on the specific characteristics of their process, considering factors such as system variability and desired update frequency.

Additionally, if detailed guidance on selecting values for `H_delta` and `H_minus` is needed, it is recommended to review the referenced article, which presents calibration and adjustment strategies for these limits to ensure optimal control chart performance.

Usage

```
plot_GICCLdown_Chart(
  alpha,
  beta,
  beta_ratio,
  H_delta,
  H_minus,
  known_alpha,
  k_l,
  delay,
  tau,
  n_I,
  n_II,
  faseI = NULL,
  faseII = NULL
)
```

Arguments

`alpha`	Shape parameter of the Gamma distribution (if `alpha_conocido = TRUE`).
`beta`	Scale parameter of the Gamma distribution.
`beta_ratio`	Ratio between `beta` and its estimation.
`H_delta`	Increment of the lower control limit.
`H_minus`	Initial lower control limit of the CUSUM chart.
`known_alpha`	Indicates whether `alpha` is known (`TRUE`) or should be estimated (`FALSE`).
`k_l`	Secondary control threshold used in the learning logic.
`delay`	Number of observations before updating `beta0_est` and `H_minus_c`.
`tau`	Time point at which the `beta` parameter changes.
`n_I`	Sample size in Phase I (if `faseI` is not provided).
`n_II`	Sample size in Phase II (if `faseII` is not provided).
`faseI`	Data sample from Phase I (numeric vector). If `NULL`, it is generated internally.
`faseII`	Data sample from Phase II (numeric vector). If `NULL`, it is generated internally.

Value

A plot showing the evolution of the downward CUSUM statistic with cautious learning, including:

- The dynamically adjusted accumulated values of the CUSUM statistic.
- Progressively updated control limits with guaranteed performance.
- A summary of the parameters used in the control chart.

Examples

```
# Option 1: Providing Phase I and Phase II data
phaseI_data <- rgamma(n = 200, shape = 1, scale = 1)
phaseII_data <- rgamma(n = 710, shape = 1, scale = 1)
plot_GICCLdown_Chart(alpha = 1, beta = 1, beta_ratio = 1/2, H_delta = 4.2433,
                     H_minus= -4.8257, known_alpha = FALSE, k_l = 0.739588,
                     delay = 25, tau = 1, n_I = 200, n_II = 700,
                     faseI = phaseI_data, faseII = phaseII_data)

# Option 2: Without providing data, the function automatically generates them
plot_GICCLdown_Chart(alpha = 1, beta = 1, beta_ratio = 1/2, H_delta = 1.6763,
                     H_minus = -4.8257, known_alpha = FALSE, k_l = 0.739588,
                     delay = 25, tau = 1, n_I = 200,
                     n_II = 710, faseI = NULL, faseII = NULL)
```

---

plot_GICCLup_Chart   *Upward CUSUM Control Chart with Cautious Learning and Guaranteed Performance*

Description

This function generates an upward CUSUM control chart for a Gamma distribution, incorporating a cautious parameter update mechanism based on guaranteed performance.

It enables dynamic process monitoring, ensuring progressive adaptation to distribution changes. This approach follows the methodology presented in the work of Madrid-Alvarez, García-Díaz, and Tercero-Gómez (2024), where a cautious learning scheme for parameter updates in CUSUM control charts applied to Gamma distributions is proposed.

The implementation captures distribution changes and adjusts the control limits to improve process variation detection.

Features::

- If the user does not provide Phase I and Phase II data, the function automatically generates them.
- If `known_alpha` = `TRUE`, `alpha` is fixed and not estimated.
- If `known_alpha` = `FALSE`, `alpha` is estimated from Phase I data.

- Includes dynamic control limits and a summary of parameters.
- Integrates a cautious learning scheme using the parameters `k_l`, `delay`, and `tau`.

Recommendations:

The parameters `k_l`, `delay`, and `tau` are part of the cautious learning mechanism of the CUSUM chart. These values enable the dynamic updating of `beta0_est` and `H_plus`, ensuring that the control chart progressively adapts to process changes, thus improving sensitivity in detecting deviations.

For proper implementation, it is recommended to reference the values proposed in:

Madrid-Alvarez, H. M., García-Díaz, J. C., & Tercero-Gómez, V. G. (2024). *A CUSUM control chart for the Gamma distribution with cautious parameter learning.* Quality Engineering, 1-23.

While these parameters have been tested and validated in the referenced article, users can adjust them based on the specific characteristics of their process, considering factors such as system variability and desired update frequency.

Additionally, if detailed guidance on selecting values for `H_delta` and `H_plus` is required, it is recommended to review the referenced article, which presents calibration and adjustment strategies for these limits to ensure optimal control chart performance.

Usage

```
plot_GICCLup_Chart(
  alpha,
  beta,
  beta_ratio,
  H_delta,
  H_plus,
  known_alpha,
  k_l,
  delay,
  tau,
  n_I,
  n_II,
  faseI = NULL,
  faseII = NULL
)
```

Arguments

`alpha`	Shape parameter of the Gamma distribution (if `known_alpha = TRUE`).
`beta`	Scale parameter of the Gamma distribution.
`beta_ratio`	Ratio between `beta` and its estimation.
`H_delta`	Increment of the upper control limit.
`H_plus`	Initial upper control limit of the CUSUM chart.
`known_alpha`	Indicates whether `alpha` is known (`TRUE`) or should be estimated (`FALSE`).
`k_l`	Secondary control threshold used in the learning logic.

delay	Number of observations before updating beta0_est and H_plus_c.
tau	Time point at which the beta parameter changes.
n_I	Sample size in Phase I (if faseI is not provided).
n_II	Sample size in Phase II (if faseII is not provided).
faseI	Data sample from Phase I (numeric vector). If NULL, it is generated internally.
faseII	Data sample from Phase II (numeric vector). If NULL, it is generated internally.

Value

A plot showing the evolution of the upward CUSUM statistic with cautious learning, including:

- The dynamically adjusted accumulated values of the CUSUM statistic.
- Progressively updated control limits with guaranteed performance.
- A summary of the parameters used in the control chart.

Examples

```
# Option 1: Providing Phase I and Phase II data
phaseI_data <- rgamma(n = 200, shape = 1, scale = 1)
phaseII_data <- rgamma(n = 710, shape = 1, scale = 1)
plot_GICCLup_Chart(alpha = 1, beta = 1, beta_ratio = 2, H_delta = 4.2433,
                   H_plus = 8.9345, known_alpha = FALSE, k_l = 2, delay = 25,
                   tau = 1,n_I = 200, n_II = 700, faseI = phaseI_data,
                   faseII = phaseII_data)

# Option 2: Without providing data, the function automatically generates them
plot_GICCLup_Chart(alpha = 1, beta = 1, beta_ratio = 2, H_delta = 2.9819,
                   H_plus = 6.5081, known_alpha = TRUE, k_l = 2, delay = 25,
                   tau = 1, n_I = 200, n_II = 710, faseI = NULL,
                   faseII = NULL)
```

---

plot_GICCL_chart2 *CUSUM Control Chart with Cautious Learning and Guaranteed Performance*

Description

This function generates a bidirectional (upward and downward) CUSUM control chart for a Gamma distribution, incorporating a cautious parameter update mechanism with guaranteed performance. Its purpose is to enhance sensitivity and precision in detecting changes in dynamic processes.

Based on the methodology presented by Madrid-Alvarez, García-Díaz, and Tercero-Gómez (2024), this implementation allows control limits to adapt according to the evolution of the process, ensuring early detection of variations while minimizing the risk of false alarms.

Features::

- If the user does not provide Phase I and Phase II data, the function automatically generates them.
- If known_alpha = TRUE, alpha is fixed and not estimated.
- If known_alpha = FALSE, alpha is estimated from Phase I data.
- Includes dynamic control limits and a summary table of parameters.
- Enables the detection of both upward and downward deviations, progressively adjusting the control limits.

Recommendations:

- The parameters k_l, delay, and tau are crucial for the learning process in the control chart. They regulate the progressive update of control limits, allowing the dynamic update of beta0_est, H_plus_c, and H_minus_c, ensuring that the control chart gradually adjusts to changes in the process. It is recommended to use reference values presented in:
  Madrid-Alvarez, H. M., García-Díaz, J. C., & Tercero-Gómez, V. G. (2024). *A CUSUM control chart for the Gamma distribution with cautious parameter learning.* Quality Engineering, 1-23.
- Similar to the parameters above, for proper selection of H_plus, H_minus, H_delta_plus, and H_delta_minus values, it is recommended to review the reference article, where detailed calibration strategies for different scenarios are presented.

Usage

```
plot_GICCL_chart2(
  alpha,
  beta,
  beta_ratio_plus,
  beta_ratio_minus,
  H_delta_plus,
  H_plus,
  H_delta_minus,
  H_minus,
  known_alpha,
  k_l,
  delay,
  tau,
  n_I,
  n_II,
  faseI = NULL,
  faseII = NULL
)
```

Arguments

alpha — Shape parameter of the Gamma distribution (if known_alpha = TRUE).

beta — Scale parameter of the Gamma distribution.

beta_ratio_plus — Ratio between beta and its estimate for upward detection.

beta_ratio_minus
: Ratio between beta and its estimate for downward detection.

H_delta_plus
: Increment of the upper control limit.

H_plus
: Initial upper limit of the CUSUM chart.

H_delta_minus
: Increment of the lower control limit.

H_minus
: Initial lower limit of the CUSUM chart.

known_alpha
: Indicates whether alpha is known (TRUE) or should be estimated (FALSE).

k_l
: Secondary control threshold used in the learning logic.

delay
: Number of observations before updating beta0_est, H_plus_c, and H_minus_c.

tau
: Time point at which the beta parameter changes.

n_I
: Sample size in Phase I (if faseI is not provided).

n_II
: Sample size in Phase II (if faseII is not provided).

faseI
: Data sample from Phase I (numeric vector). If NULL, it is generated internally.

faseII
: Data sample from Phase II (numeric vector). If NULL, it is generated internally.

Value

A plot showing the evolution of the CUSUM statistic with cautious learning, including:

- Dynamically adjusted accumulated values of the CUSUM statistic.
- Progressively updated control limits with guaranteed performance.
- A summary of the parameters used in the control chart.

Examples

```
# Option 1: Automatically generated data
plot_GICCL_chart2(alpha = 1, beta = 1,
                 beta_ratio_plus = 2, beta_ratio_minus = 0.5,
                 H_delta_plus = 3.0, H_plus = 6.5,
                 H_delta_minus = 2.0, H_minus = -5.0,
                 known_alpha = TRUE, k_l = 2, delay = 25, tau = 1,
                 n_I = 200, n_II = 700,
                 faseI = NULL, faseII = NULL)

# Option 2: User-provided data
datos_faseI <- rgamma(n = 200, shape = 1, scale = 1)
datos_faseII <- rgamma(n = 700, shape = 1, scale = 1)
plot_GICCL_chart2(alpha = 1, beta = 1,
                 beta_ratio_plus = 2, beta_ratio_minus = 0.5,
                 H_delta_plus = 3.0, H_plus = 6.5,
                 H_delta_minus = 2.0, H_minus = -5.0,
                 known_alpha = FALSE, k_l = 2, delay = 25, tau = 1,
                 n_I = 200, n_II = 700,
                 faseI = datos_faseI, faseII = datos_faseII)
```

`plot_GICCup_chart` *Upward CUSUM Control Chart for Gamma Distribution with Guaranteed Performance*

Description

This function generates an upward CUSUM control chart for a Gamma distribution, displaying the evolution of the CUSUM statistic, control limits, and a summary of the parameters.

Based on the approach presented by Madrid-Alvarez, García-Díaz, and Tercero-Gómez (2024), this implementation enables the evaluation and visualization of the monitored process using a CUSUM chart adapted to Gamma distributions with guaranteed performance.

Specifically, the library incorporates a Monte Carlo model for simulating the control chart behavior, allowing the Gamma distribution to be estimated in Phase I or using predefined values. Additionally, it provides a clear graphical representation of the CUSUM statistic's evolution, ensuring proper calibration and process control.

Recommendations:

To check specific values for `H_delta` and `H_plus`, it is recommended to review the reference article: Madrid-Alvarez, H. M., García-Díaz, J. C., & Tercero-Gómez, V. G. (2024). **A CUSUM control chart for gamma distribution with guaranteed performance**. Quality and Reliability Engineering International, 40(3), 1279-1301.

Features::

- Based on a Monte Carlo model.
- Estimates the Gamma distribution in Phase I or uses predefined values.
- Plots the accumulated values of the CUSUM statistic with guaranteed performance.
- Includes control limits and a summary table.

Usage

```
plot_GICCup_chart(
  alpha,
  beta,
  beta_ratio,
  H_delta,
  H_plus,
  n_I,
  n_II,
  faseI = NULL,
  faseII = NULL,
  known_alpha
)
```

Arguments

`alpha`	Shape parameter of the Gamma distribution.
`beta`	Scale parameter of the Gamma distribution.
`beta_ratio`	Ratio between beta and its estimation.
`H_delta`	Increment of the upper GIC limit.
`H_plus`	Initial upper limit of the CUSUM chart.
`n_I`	Sample size in Phase I (if `faseI` is not provided).
`n_II`	Sample size in Phase II (if `faseII` is not provided).
`faseI`	Sample data from Phase I (numeric vector). If `NULL`, it is generated with `rgamma()`.
`faseII`	Sample data from Phase II (numeric vector). If `NULL`, it is generated with `rgamma()`.
`known_alpha`	If `TRUE`, a known `alpha` is used; if `FALSE`, it is estimated.

Value

A plot displaying the evolution of the upward CUSUM statistic, including:

- The accumulated values of the CUSUM statistic.
- Control limits with guaranteed performance.
- A summary of the parameters used in the control chart.

Examples

```
# Option 1: Automatically generate data with defined sample sizes
plot_GICCup_chart(
                  alpha = 1, beta = 1, beta_ratio = 2, H_delta = 0,
                  H_plus = 5.16, n_I = 100, n_II = 200, faseI = NULL,
                  faseII = NULL, known_alpha = TRUE
                  )

# Option 2: Use custom data
phaseI_data <- rgamma(n = 100, shape = 1, scale = 1)
phaseII_data <- rgamma(n = 200, shape = 1, scale = 1)
plot_GICCup_chart(
                  alpha = 1, beta = 1, beta_ratio = 2, H_delta = 2.9693,
                  H_plus = 6.5081, n_I = 100, n_II = 200,
                  faseI = phaseI_data, faseII = phaseII_data,
                  known_alpha = TRUE
                  )
```

`plot_GICC_chart2` *CUSUM Control Chart for Gamma Distribution with Guaranteed Performance*

Description

This function generates a bidirectional (upward and downward) CUSUM control chart for a Gamma distribution, allowing the monitoring of the evolution of the CUSUM statistic while ensuring optimal performance in detecting process changes.

Based on the methodology proposed by Madrid-Alvarez, García-Díaz, and Tercero-Gómez (2024), this implementation employs a Monte Carlo-based approach to estimate the Gamma distribution parameters and determine control limits with precise calibration. The function enables visualization of process evolution and the detection of deviations with reduced risk of false alarms.

Features::

- Implements Monte Carlo simulations for control chart calibration.
- Allows the use of known Gamma distribution values or estimation in Phase I.
- Provides a graphical representation of the CUSUM statistic evolution with guaranteed performance.
- Includes control limits and a legend with key configuration details of the control chart.

For additional details on selecting parameters `H_plus`, `H_minus`, `H_delta_plus`, and `H_delta_minus`, as well as calibration strategies, it is recommended to consult the reference article:

Madrid-Alvarez, H. M., García-Díaz, J. C., & Tercero-Gómez, V. G. (2024). *A CUSUM control chart for gamma distribution with guaranteed performance.* Quality and Reliability Engineering International, 40(3), 1279-1301.

Usage

```
plot_GICC_chart2(
  alpha,
  beta,
  beta_ratio_plus,
  beta_ratio_minus,
  H_delta_plus,
  H_plus,
  H_delta_minus,
  H_minus,
  n_I,
  n_II,
  faseI = NULL,
  faseII = NULL,
  known_alpha
)
```

Arguments

- `alpha` Shape parameter of the Gamma distribution.
- `beta` Scale parameter of the Gamma distribution.
- `beta_ratio_plus` Ratio between beta and its estimation for upward detection.
- `beta_ratio_minus` Ratio between beta and its estimation for downward detection.
- `H_delta_plus` Increment of the upper GIC limit.
- `H_plus` Initial upper limit of the CUSUM chart.
- `H_delta_minus` Increment of the lower GIC limit.
- `H_minus` Initial lower limit of the CUSUM chart.
- `n_I` Sample size in Phase I (if `faseI` is not provided).
- `n_II` Sample size in Phase II (if `faseII` is not provided).
- `faseI` Sample data from Phase I (numeric vector). If `NULL`, it is generated with `rgamma()`.
- `faseII` Sample data from Phase II (numeric vector). If `NULL`, it is generated with `rgamma()`.
- `known_alpha` If `TRUE`, a known `alpha` is used; if `FALSE`, it is estimated.

Value

A plot showing the evolution of the CUSUM statistic for a Gamma distribution with guaranteed performance, including:

- The accumulated values of the CUSUM statistic.
- Control limits with guaranteed performance.
- A summary of the parameters used in the control chart.

Examples

```
# Option 1: Automatically generate data with predefined sample sizes
plot_GICC_chart2(alpha = 1, beta = 1, beta_ratio_plus = 2,
                      beta_ratio_minus = 0.5,H_delta_plus = 2.0,
                      H_plus = 5.0, H_delta_minus = 1.5, H_minus = -4.5,
                      n_I = 100, n_II = 200, faseI = NULL,
                      faseII = NULL, known_alpha = TRUE
                     )

# Option 2: Use custom data
phaseI_data <- rgamma(n = 100, shape = 1, scale = 1)
phaseII_data <- rgamma(n = 200, shape = 1, scale = 1)
plot_GICC_chart2(alpha = 1, beta = 1, beta_ratio_plus = 2,
                 beta_ratio_minus = 0.5, H_delta_plus = 2.0, H_plus = 5.0,
                 H_delta_minus = 1.5, H_minus = -4.5, n_I = 100, n_II = 200,
                 faseI = phaseI_data, faseII = phaseII_data,
                 known_alpha = TRUE
                 )
```

Index