Package ‘CTxCC’

August 28, 2025

Type Package

Title Multivariate Normal Mean Monitoring Through Critical-to-X Control Chart

Version 0.4.0

Maintainer Diana Barraza-Barraza <diana.barraza@ujed.mx>

Description A comprehensive set of functions designed for multivariate mean monitoring using the Critical-to-X Control Chart. These functions enable the determination of optimal control limits based on a specified in-control Average Run Length (ARL), the calculation of out-of-control ARL for a given control limit, and post-signal analysis to identify the specific variable responsible for a detected shift in the mean. This suite of tools provides robust support for precise and effective process monitoring and analysis.

SystemRequirements Intel MKL (optional for optimized BLAS/LAPACK performance)

License GPL (>= 2)

Encoding UTF-8

Depends combinat, matrixcalc, stats, mvtnorm, expm, CompQuadForm, ggplot2

NeedsCompilation no

Author Burcu Aytaçoğlu [aut] (burcuaytacoglu@gmail.com),
Diana Barraza-Barraza [aut, cre] (diana.barraza@ujed.mx),
Víctor G. Tercero-Gómez [aut] (victor.tercero@tec.mx),
A. Eduardo Cordero-Franco [aut] (lalo.cordero@gmail.com)

Repository CRAN

Date/Publication 2025-08-27 23:20:02 UTC

Contents

`C2.Contribution` *Contribution of variable z.var to C^2*

Description

Returns contribution of variable z.var to C^2, even if there are no previous variables in the model

Usage

```
C2.Contribution(z, mean0, W, R, x.var, z.var = NULL)
```

Arguments

`z`	observation vector, kx1, where z[x.var,] correspond to variables already in the model
`mean0`	Mean vector for multivariate random vector under the null hypothesis. Dimensions: kx1
`W`	matrix of variables weigths, kxk
`R`	correlation matrix, kxk
`x.var`	vector indicating variables already present in the model. length: k-1
`z.var`	scalar indicating variables to be included. Defaults to NULL, indicating there are no previous variables in the model

Value

C2.k.extra, scalar containing the contribution of variable z to C_k

Author(s)

Dr. Burcu Aytaçoğlu (burcuaytacoglu@gmail.com) Dr. Diana Barraza-Barraza (diana.barraza@ujed.mx), Dr. Víctor G. Tercero-Gómez (victor.tercero@tec.mx), Dr. A. Eduardo Cordero-Franco (lalo.cordero@gmail.com),

References

Paper

Examples

```
k<-3
sigma0 = matrix(diag(rep(1,k)),ncol = k)
mu0 = matrix(c(0,0,0), ncol = 1)
Weights = diag(c(0.5, 0.25,0.25))

library(mvtnorm)
set.seed(1000)
X = matrix(ncol= 1, data = rmvnorm(n = 1, mean = mu0, sigma = sigma0))
Z = (X - mu0)/sqrt(as.numeric(diag(sigma0)))
Corr<-get.R(Sigma0 = sigma0)

C2.Contribution(z = Z, W = Weights, R = Corr, x.var = 1:2, z.var = 3)
```

C2.DecisionLimit *Conditional decision limit for z, given x already in model*

Description

Calculates the conditional decision limit for z, given x already in model, using the exact distribution for the conditional contribution of z to C_k

Usage

```
C2.DecisionLimit(z, mu.C, R.C, A, x.var, alpha)
```

Arguments

`z`	observation vector, kx1, where z[x.var,] correspond to variables already in the model
`mu.C`	scalar, conditional mean for z given x
`R.C`	scalar, conditional covariance for z given x
`A`	list containing matrix decomposition of A, preferably, obtained from function decomposeA
`x.var`	vector indicating variables already present in the model. length: k-1
`alpha`	confidence level for decision limit

Details

Proposition Distribution of a C^2 contribution from Paper Criticality Assessment for Enhanced Multivariate Process Monitoring

Value

conditionalCL, conditional decision limit for z's contribution to C_k

Author(s)

Dr. Burcu Aytaçoğlu (burcuaytacoglu@gmail.com) Dr. Diana Barraza-Barraza (diana.barraza@ujed.mx), Dr. Víctor G. Tercero-Gómez (victor.tercero@tec.mx), Dr. A. Eduardo Cordero-Franco (lalo.cordero@gmail.com),

References

Paper

Examples

```
k<-3
sigma0 = matrix(diag(rep(1,k)),ncol = k)
mu0 = matrix(c(0,0,0), ncol = 1)
Weights = diag(c(0.5, 0.25,0.25))

library(mvtnorm)
set.seed(1000)
X = matrix(ncol= 1, data = rmvnorm(n = 1, mean = mu0, sigma = sigma0))
Z = (X - mu0)/sqrt(as.numeric(diag(sigma0)))
Corr<-get.R(Sigma0 = sigma0)

A<-decomposeA(W = Weights, R = Corr, x.var = 1:2, z.var = 3)

Par<-zConditionalParameters(mean0 = mu0, R0 = Corr, z = Z, x.var = 1:2, z.var = 3)
C2.DecisionLimit(z = Z, mu.C = Par$muC, R.C = Par$RC, A = A, x.var = 1:2, alpha = 0.95)
```

`c2decomp` *Contribution to C^2 for all variables*

Description

Returns a matrix with values for C^2_1 and C^2_k|C^2_k-1,C^2_k-2, ..., C^2_1, k=2,3, 4... for all possible permutations among k variables

Usage

```
c2decomp(z, W, R)
```

Arguments

`z`	observation vector, kx1
`W`	matrix of variables weigths, kxk
`R`	correlation matrix, kxk

Value

Data frame where, the first k columns correspond to variable that entred the model first, second... k-th. The following (k+1) to 2*k columns contain the conditional contribution of the variable. The last column contains the sum of all contributions, meaning C^2_k

Author(s)

Dr. Burcu Aytaçoğlu (burcuaytacoglu@gmail.com) Dr. Diana Barraza-Barraza (diana.barraza@ujed.mx), Dr. Víctor G. Tercero-Gómez (victor.tercero@tec.mx), Dr. A. Eduardo Cordero-Franco (lalo.cordero@gmail.com),

References

Paper

Examples

```
k<-3
sigma0 = matrix(diag(rep(1,k)),ncol = k)
mu0 = matrix(c(0,0,0), ncol = 1)
Weights = diag(c(0.5, 0.25,0.25))

library(mvtnorm)
set.seed(1000)
X = matrix(ncol= 1, data = rmvnorm(n = 1, mean = mu0, sigma = sigma0))
Z = (X - mu0)/sqrt(as.numeric(diag(sigma0)))
Corr<-get.R(Sigma0 = sigma0)

c2decomp(z = Z, W = Weights, R = Corr)
```

---

contributionplot *Mean Decomposition Pareto Plot*

Description

Plots the mean contribution of each variable to the C^2 statistic obtained from the C2.allPerms function. The graph follows a Pareto-style layout, ordering variables by the magnitude and percentage of their contribution. It also indicates the number of permutations in which each variable's contribution is statistically significant.

Usage

```
contributionplot(varnames=NULL, X, alpha, title=NULL)
```

Arguments

varnames=NULL, X, alpha, title=NULL

varnames	Defaults to NULL. A vector containing variable names
X	list generated by C2.allPerms function, containing each permutation contribution and respective p-value
alpha	an alpha value used to define the variables significance by compairing their p-values to alpha
title	plot title

Value

A Pareto plot showing the mean contribution of each variable to the C^2 statistic. The plot includes information on the number of permutations, the mean and percentage contribution of each variable, and the number of permutations in which the variable is significant, based on its p-value compared to the specified alpha level.

Author(s)

Dr. Burcu Aytaçoğlu (<burcuaytacoglu@gmail.com>), Dr. Diana Barraza-Barraza (<diana.barraza@ujed.mx>), Dr. Víctor G. Tercero-Gómez (<victor.tercero@tec.mx>), Dr. A. Eduardo Cordero-Franco (<lalo.cordero@gmail.com>)

References

Paper (pending complete bibliographic reference when available).

Examples

```
k<-3
sigma0 = matrix(diag(rep(1,k)),ncol = k)
mu0 = matrix(c(0,0,0), ncol = 1)
Weights = diag(c(0.5, 0.25,0.25))

library(mvtnorm)
set.seed(1000)
X = matrix(ncol= 1, data = rmvnorm(n = 1, mean = mu0, sigma = sigma0))
Z = (X - mu0)/sqrt(as.numeric(diag(sigma0)))
Corr<-get.R(Sigma0 = sigma0)

X<-c2decomp(z = Z, W = Weights, R = Corr)
contributionplot(varnames = paste0("x",1:3), X = X, alpha = .05, title = "Z")
```

CTXoptim *Optimum Design of a CTX Control Chart*

Description

Computes the optimal weight vector and control limit for a CTX control chart that minimizes the overall out-of-control Average Run Length (ARL) for a given set of shift vectors. The optimization can be performed using either a single-start or a multistart approach, depending on the number of variables in the dataset. For the single-start case, the BFGS method is used, while in the multistart case, the optimization is performed using either BFGS or Nelder–Mead.

Usage

```
CTXoptim(delta, R, alpha,
n_starts, method_primary,
method_secondary, scale_sum,jitter_sd, seed)
```

Arguments

`delta`	Numeric. Shift size or shift vector considered in the optimization.
`R`	Numeric. Correlation matrix or covariance structure used in the CTX control chart.
`alpha`	Numeric. Significance level (Type I error rate) for the control chart.
`n_starts`	Integer. Number of starting points for the optimization. Default is `30`.
`method_primary`	Character. Optimization method for the single-start case. Default is `"BFGS"`.
`method_secondary`	Character. Optimization method for the multistart case (alternative to primary). Default is `"Nelder-Mead"`.
`scale_sum`	Numeric. Scaling constant applied to the weight vector. Default is `1`.
`jitter_sd`	Numeric. Standard deviation of the jitter added when generating starting points in multistart optimization. Default is `0.3`.
`seed`	Integer. Random seed for reproducibility in multistart optimization. Default is `123`.

Details

The optimization is performed over proxy weights (log-transformed) to ensure positivity, and the weights are normalized to sum to one. The user can choose between single-start and multistart optimization, depending on the number of variables (and out-of-control ARLs) to be monitored. A quasi-Newton method (`BFGS`) is used to minimize the sum of out-of-control ARLs across the specified shift scenarios. In the multistart case, if BFGS fails, the Nelder–Mead algorithm is applied instead. The control limit is computed to satisfy the desired in-control ARL, with `alpha = 1/ARL0`.

Value

weights	The optimized weight vector. Sum up to scale_sum
control_limit	The computed control limit.
objective	The optimal value of the objective function.
method	The optimization method used.
n_starts	The number of starting points used in the optimization.

Author(s)

Dr. Burcu Aytaçoğlu (<burcuaytacoglu@gmail.com>), Dr. Diana Barraza-Barraza (<diana.barraza@ujed.mx>), Dr. Víctor G. Tercero-Gómez (<victor.tercero@tec.mx>), Dr. A. Eduardo Cordero-Franco (<lalo.cordero@gmail.com>)

References

Paper (pending complete bibliographic reference when available).

Examples

```
## Not run:
# Define shift scenarios
delta1 <- c(1, 0.5)
delta <- list(delta1)
rho <- 0.75

 # Correlation matrix and false alarm rate
 R <- diag(2); R[1,2] <- R[2,1] <- rho
alpha <- 1/200

 # Compute optimal weights and control limit
 CTXoptim(delta = delta, R = R, alpha = alpha,
                       n_starts = 30, scale_sum = 1, seed = 123)

## End(Not run)
```

---

decomposeA *Calculation and decomposition of matrix A*

Description

Decomposition of matrix A, required in Proposition 4.3. Decomoposition given by equation 41

Usage

```
decomposeA(W, R, x.var, z.var)
```

Arguments

W	diagonal matrix containing the corresponding weigth for each monitored variable. Dimensions kxk
R	correlation matrix for monitores variables, kxk
x.var	vector indicating variables already present in the model. length: k-1.
z.var	scalar indicating variables to be included.

Details

Note that length(z.var) + length(x.var) = k

Value

Returns decomposition of matrix A according to Equation 41 in paper.

Author(s)

Dr. Burcu Aytaçoğlu (burcuaytacoglu@gmail.com) Dr. Diana Barraza-Barraza (diana.barraza@ujed.mx),
Dr. Víctor G. Tercero-Gómez (victor.tercero@tec.mx), Dr. A. Eduardo Cordero-Franco (lalo.cordero@gmail.com),

References

Paper

Examples

```
k<-6 # variables
B<-matrix(runif(n = k*k),ncol= k)### creating random matrix for sigma0
sigma0 <- B%*%t(B)
R<-get.R(sigma0)
Weights = diag(rep(1/k,k))
decomposeA(W = Weights, R = R, x.var = 1:5, z.var = 6)
```

---

get.R *Get Correlation matrix from a Covariance matrix*

Description

Returns a correlation matrix from a variance-covariance matrix

Usage

```
get.R(Sigma0)
```

Arguments

Sigma0	variance-covariance matrix of dimensions kxk

Value

R	correlation matrix correspondig to Sigma0

Author(s)

Dr. Burcu Aytaçoğlu (burcuaytacoglu@gmail.com) Dr. Diana Barraza-Barraza (diana.barraza@ujed.mx), Dr. Víctor G. Tercero-Gómez (victor.tercero@tec.mx), Dr. A. Eduardo Cordero-Franco (lalo.cordero@gmail.com),

References

Paper

Examples

```
k<-6 # variables
B<-matrix(runif(n = k*k),ncol= k)### creating random matrix for sigma
sigma = B%*%t(B)
get.R(Sigma0=sigma)
```

---

ms.optim — *Multistart Weight Optimization for Design of a CTX Control Chart*

Description

Computes the optimal weight vector and control limit for a CTX control chart that minimizes the overall out-of-control Average Run Length (ARL) across a given set of shift scenarios. Optimization is performed using a multistart approach. A quasi-Newton method ("BFGS") is used by default, with the Nelder–Mead algorithm as a backup in multistart optimization.

Usage

```
ms.optim(delta, R, alpha, n_starts,
method_primary, method_secondary,
scale_sum,jitter_sd, seed)
```

Arguments

delta	List. A set of shift vectors to be evaluated in the optimization.
R	Numeric matrix. Correlation (or covariance) matrix for the CTX control chart.
alpha	Numeric. Significance level for the in-control ARL, where alpha = 1/ARL0.
n_starts	Integer. Number of random starting points for multistart optimization.
method_primary	Character. Primary optimization method.
method_secondary	Character. Secondary (backup) optimization method used if the primary method fails.
scale_sum	Numeric. Scaling constant applied to normalize the weight vector.

`jitter_sd`	Numeric. Standard deviation of Gaussian noise used to jitter starting points in multistart optimization.
`seed`	Integer. Random seed for reproducibility.

Details

The optimization is performed over log-transformed proxy weights to ensure positivity, and the resulting weights are normalized to sum to `scale_sum`. The objective function minimizes the sum of out-of-control ARLs across all specified shift scenarios. The control limit is computed to satisfy the desired in-control ARL, given by `alpha = 1/ARL0`.

Value

`weights`	Optimized weight vector.
`control_limit`	Computed control limit.
`objective`	Optimal value of the objective function.
`method`	Optimization method used.
`n_starts`	Number of starting points used in the optimization.

Author(s)

Dr. Burcu Aytaçoğlu (<burcuaytacoglu@gmail.com>), Dr. Diana Barraza-Barraza (<diana.barraza@ujed.mx>), Dr. Víctor G. Tercero-Gómez (<victor.tercero@tec.mx>), Dr. A. Eduardo Cordero-Franco (<lalo.cordero@gmail.com>)

References

Paper (pending complete bibliographic reference when available).

Examples

```
## Not run:
# Define shift scenarios

R <- diag(3)
delta <- list(c(0.5, 0, 0), c(0, 0.5, 0), c(0, 0, 0.5))

# Run multi-start optimization
res1 <- ms.optim(delta = delta, R = R, alpha = 0.005, n_starts = 10)

# Run multistart optimization with jittered initial points
res2 <- ms.optim(delta = delta, R = R, alpha = 0.005, n_starts = 5,
                 method_primary = "BFGS", method_secondary = "Nelder-Mead")

## End(Not run)
```

SimulatedDistributionC2

Distribution for C2, through simulation of its values

Description

Simulates s instances of C^2_k given 1 to k-1 variables are already in the model. Obtains the quantile indicated by alpha

Usage

```
SimulatedDistributionC2(z, R.C, mu.C, W, R, A, x.var, z.var, alpha, s)
```

Arguments

z	observation vector, kx1
R.C	scalar, conditional covariance for z given x
mu.C	scalar, conditional mean for z given x
W	matrix of variables weigths, kxk
R	correlation matrix, kxk
A	list containing matrix decomposition of A, preferably, obtained from function decomposeA
x.var	vector indicating variables already present in the model. length: k-1
z.var	scalar indicating variable to be included
alpha	quantile(s) of the distribution
s	scalar indicating amount of simulations

Value

Quantile(s) of the simulated distribution

Author(s)

Dr. Burcu Aytaçoğlu (burcuaytacoglu@gmail.com) Dr. Diana Barraza-Barraza (diana.barraza@ujed.mx),
Dr. Víctor G. Tercero-Gómez (victor.tercero@tec.mx), Dr. A. Eduardo Cordero-Franco (lalo.cordero@gmail.com),

References

Paper

Examples

```
k<-3
sigma0 = matrix(diag(rep(1,k)),ncol = k)
mu0 = matrix(c(0,0,0), ncol = 1)
Weights = diag(c(0.5, 0.25,0.25))

library(mvtnorm)
set.seed(1000)
X = matrix(ncol= 1, data = rmvnorm(n = 1, mean = mu0, sigma = sigma0))
Z = (X - mu0)/sqrt(as.numeric(diag(sigma0)))
Corr<-get.R(Sigma0 = sigma0)

A<-decomposeA(W = Weights, R = Corr, x.var = 1:2, z.var = 3)

Par<-zConditionalParameters(mean0 = mu0, R0 = Corr, z = Z, x.var = 1:2, z.var = 3)
SimulatedDistributionC2(z = Z, R.C = Par$RC, mu.C = Par$muC, W = Weights, R = Corr,
                        A = A, x.var = 1:2, z.var = Z, alpha = 0.95, s = 1000 )
```

---

ss.optim — *Single-Start Weight Optimization for Design of a CTX Control Chart*

Description

Computes the optimal weight vector and control limit for a CTX control chart that minimizes the overall out-of-control Average Run Length (ARL) for a given set of shift vectors. The optimization can be performed using either a single-start or a multistart approach, depending on the number of variables in the dataset. For the single-start case, the BFGS method is used, while in the multistart case, the optimization is performed using either BFGS or Nelder–Mead.

Usage

```
ss.optim(delta, R, alpha)
```

Arguments

`delta`	list of numeric vectors, each representing a mean shift scenario to be optimized for (out-of-control cases).
`R`	A positive-definite correlation matrix for the monitored variables.
`alpha`	Target in-control false alarm rate (e.g., 1/500) corresponds to an in-control ARL of 500).

Details

The optimization is performed over proxy weights (log-transformed) to ensure positivity. These weights are normalized to sum to one, and a quasi-Newton method (‘BFGS‘) is used to minimize the sum of out-of-control ARLs computed for each specified shift scenario. The control limit is computed to satisfy the desired in-control ARL (via `alpha` = 1/ARL0).

Value

`weights` Numeric vector of optimal weights that sum to one.

`control_limit` Numeric value of the CTX control chart limit that satisfies the specified alpha

.

Author(s)

Dr. Burcu Aytaçoğlu (<burcuaytacoglu@gmail.com>), Dr. Diana Barraza-Barraza (<diana.barraza@ujed.mx>), Dr. Víctor G. Tercero-Gómez (<victor.tercero@tec.mx>), Dr. A. Eduardo Cordero-Franco (<lalo.cordero@gmail.com>)

References

Paper (pending complete bibliographic reference when available).

Examples

```
## Not run:
# Define shift scenarios
delta1 <- c(1, 0, 0)
delta2 <- c(0, 3, 0)
delta3 <- c(0, 0, 2)
delta <- list(delta1, delta2, delta3)

 # Correlation matrix and false alarm rate
 R <- diag(3)
 alpha <- 1/500

 # Compute optimal weights and control limit
 optim_ss(delta = delta, R = R, alpha = alpha)

## End(Not run)
```

`wChisq.arl` *Compute ARLs of Weighted Chi-Squared control charts for monitoring multivariate normal mean.*

Description

Computation of the Average Run Length (ARL) for a Weighted Chi-Squared control chart for a given mean vector, delta, correlation matrix, R, control limit, h, and the vector of weights, w. The mean vector, delta, is defined in Propositon 4.2 from Paper Criticality Assessment for Enhanced Multivariate Process Monitoring.

Usage

```
wChisq.arl(delta, R, h,  w)
```

Arguments

`delta`	Vector of values representing the change in the mean for each variable, 1xk
`R`	correlation matrix, kxk
`h`	Control limit of Weighted Chi-Squared Control chart
`w`	vector of weigths, 1xk

Value

`arl`	Average Run Length (ARL) for a Weighted Chi-Squared control chart for a given mean vector

Author(s)

Dr. Burcu Aytaçoğlu (burcuaytacoglu@gmail.com) Dr. Diana Barraza-Barraza (diana.barraza@ujed.mx),
Dr. Víctor G. Tercero-Gómez (victor.tercero@tec.mx), Dr. A. Eduardo Cordero-Franco (lalo.cordero@gmail.com),

References

Paper

Examples

```
#Table 1 in the Paper Criticality Assessment for Enhanced Multivariate Process Monitoring.
delta <- c(0.5, 0.5)    # mean vector (change vector)
R <- diag(2)     # correlation matrix
h <- 2.649506   # Control limit
w <- c(0.50153, 0.49847) # vector of weights
wChisq.arl(delta, R, h,  w)
```

`wChisq.CLim` *Compute control limit of Weighted Chi-Squared control charts for monitoring multivariate normal mean.*

Description

Computation of a control limit of the Weighted Chi-Squared control chart for a given vector of weights, w, correlation matrix, R, and the false alarm rate, alpha.

Usage

```
wChisq.CLim(w,R,alpha)
```

Arguments

`w`	vector of weigths, 1xk
`R`	correlation matrix, kxk
`alpha`	false alarm rate

Value

`ContLim` control limit of the Weighted Chi-Squared control chart

Author(s)

Dr. Burcu Aytaçoğlu (burcuaytacoglu@gmail.com) Dr. Diana Barraza-Barraza (diana.barraza@ujed.mx), Dr. Víctor G. Tercero-Gómez (victor.tercero@tec.mx), Dr. A. Eduardo Cordero-Franco (lalo.cordero@gmail.com),

References

Paper

Examples

```
# Table 1 in the Paper Criticality Assessment for Enhanced Multivariate Process Monitoring.

w <- c(0.29836,0.70164)  #vector of weights
R <- diag(2)
alpha <- 0.005
wChisq.CLim(w,R,alpha)

w <- c(0.23912,0.76088)  #vector of weights
R <- diag(2)
R[1,2] <- R[2,1] <- 0.25
alpha <- 0.005
wChisq.CLim(w,R,alpha)
```

---

zConditionalParameters

Conditional parameters for z, given x

Description

This function calculates and returns conditional parameters for z, given x are being already considered in the model

Usage

```
zConditionalParameters(mean0, R0, z, x.var, z.var)
```

Arguments

`mean0`	Mean vector for multivariate random vector under the null hypothesis. Dimensions: kx1
`R0`	Correlations matrix for multivariate random vector under the null hypothesis. Dimensions kxk
`z`	vector of random observation. Dimensions kx1

x.var	Elements of z that are already considered in the model
z.var	element of z whose contribution to C_k|C_k-1,C_k-2,...,C_1 is going to be calculated

Value

A list containing

muC	conditional mean for z
RC	Conditional variance for z

Author(s)

Dr. Burcu Aytaçoğlu (burcuaytacoglu@gmail.com) Dr. Diana Barraza-Barraza (diana.barraza@ujed.mx),
Dr. Víctor G. Tercero-Gómez (victor.tercero@tec.mx), Dr. A. Eduardo Cordero-Franco (lalo.cordero@gmail.com),

References

Paper

Examples

```
  k<-3
sigma0 = matrix(diag(rep(1,k)),ncol = k)
mu0 = matrix(c(0,0,0), ncol = 1)
Weights = diag(c(0.5, 0.25,0.25))

library(mvtnorm)
set.seed(1000)
X = matrix(ncol= 1, data = rmvnorm(n = 1, mean = mu0, sigma = sigma0))
Z = (X - mu0)/sqrt(as.numeric(diag(sigma0)))
Corr<-get.R(Sigma0 = sigma0)

zConditionalParameters(mean0 = mu0, R0 = Corr, z = Z, x.var = 1:2, z.var = 3)
```

Index