Package ‘AHPtools’

June 20, 2025

Type Package

Title Consistency in the Analytic Hierarchy Process

Version 1.0.0

URL https://CRAN.R-project.org/package=AHPtools

Description A Swiss Army knife of utility functions for users of the Analytic Hierarchy Process (AHP) which will help you to assess the consistency of a PCM as well as to improve its consistency ratio, to compute the sensitivity of a PCM, create a logical (as distinct from a random PCM) from preferences provided for the alternatives, and a function that helps evaluate the actual consistency of a PCM based on objective, fair bench marking. The various functions in the toolkit additionally provide the flexibility to users to specify only the upper triangular comparison ratios of the PCM in order to performs its assigned task.

License GPL-3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Depends R (>= 3.5.0)

Imports readxl, data.tree

VignetteBuilder knitr

Encoding UTF-8

Config/testthat/edition 3

Language en-US

RoxygenNote 7.3.2

NeedsCompilation no

Repository CRAN

Maintainer Amarnath Bose <`amarnath.bose@gmail.com`>

Date/Publication 2025-06-20 02:20:15 UTC

Author Amarnath Bose [aut, cre] (ORCID:
<https://orcid.org/0000-0003-1560-6745>)

Contents

`AHPweights` *Compute weights for Alternatives and lowest level sub criteria in AHP responses*

Description

This function reads an Excel file with two required Sheets, viz. Sheet 1: for the AHP structure, with three columns as follows: Sheet 2: for the upper triangular elements of the PCMs that are part of the AHP hierarchy

This returns a list of two values: (1) a printable AHP tree excluding the alternatives, if any (2) the list of weights for the lowest level subcriteria, and weights of alternatives if exists

Usage

```
AHPweights(ExcelPath, AHPsheet, PCMsheet)
```

Arguments

`ExcelPath`	for the Excel file containing the AHP structure and the required PCMs
`AHPsheet`	for the AHP structure, with three required columns, viz. Column 1: Node: the node names for all nodes that have child nodes Column 2: Parent: the parent node for the Node in Column 1 Column 3: Children_Ordered: the child nodes for the Node in Column 1. these are comma separated strings, and correspond to the ordered upper triangular elements of the PCM in Sheet 2
`PCMsheet`	for the PCMs that are part of the AHP. The upper triangular matrix elements are provided for each PCM, so that a nxn PCM has n(n-1)/2 entries. These entries have column names starting with the AHP node name with respect to which the child elements are being compared, followed by a dot (.) and a sequence of numbers from 1 to n(n-1)/2 for the PCM elements

Details

For an overview and examples, please see the associated vignette: ‘vignette("AHPweights", package = "AHPtools")‘

Value

A list of two items, (i) AHPtree which is a printable tree object constructed from the user-provided AHP structure (ii) AHPresult the list of weights for the lowest level subcriteria, and weights of alternatives if exists

Examples

```
file <- system.file("extdata", "example_transport.xlsx", package = "AHPtools")
results <- AHPweights(file, "ahp", "pcm")
print(results)
```

`consEval` *Find consistency of a PCM based on Preference Reversals*

Description

This function finds all triad based preference reversals for a PCM. #' Triads are subsets of 3 elements chosen from the 'n' alternatives of an order-n PCM. A triad reversal is said to occur if any two elements of the order-3 PCM show a reversal in preference with the corresponding elements of the full eigenvector.

This returns a list of values related to triad Preference Reversal consistency. The fourth item of the list is a data frame of triads where reversals are seen, with the logit Consistency probability measure based on them, the proportion of reversals, the maximum reversal and the data frame with the details.

Usage

```
consEval(pcm)
```

Arguments

`pcm`	A pairwise comparison matrix

Value

A list of four elements, logitConsistency = the probability that the PCM is consistent, prop3Rev = the proportion of triad-based preference reversals for the PCM, max3Rev = the maximum triad-based preference reversal for the PCM, triadsData = a data frame with 8 columns, providing the full data of preference reversals (1) triadE1 alternative 1 in the triad; e.g. a4 for the fourth alternative (2) triadE2 alternative 2 in the triad (3) triadE3 alternative 3 in the triad (4) pref3Rev measure of the intensity of preference reversal for the particular triad (5) pcmWeightE1 eigen weight of alternative triadE1 from the entire eigenvector (6) pcmWeightE2 eigen weight of alternative triadE2 from the entire eigenvector (7) triadWeightE1 eigen weight of alternative triadE1 from the order-3 sub matrix (8) triadWeightE2 eigen weight of alternative triadE2 from the order-3 sub matrix

Examples

```
pcm1 <- matrix(c(1,1,2,1,2,2, 1,1,1,1/3,1,1, 1/2,1,1,1,1,1, 1,3,1,1,2,1,
                 1/2,1,1,1/2,1,1/4, 1/2,1,1,1,4,1), nrow=6, byrow=TRUE)
cons1 <- consEval(pcm1)
cons1
pcm2 <- matrix(c(1,1/6,1/5,1/2,1/6,1/3,1/3,1/8,  6,1,1,3,1,1,2,1,
                 5,1,1,3,1/3,1,2,1/2, 2,1/3,1/3,1,1/5,1/2,1/2,1/4,
                 6,1,3,5,1,1,2,1,  3,1,1,2,1,1,1,1/3,
                 3,1/2,1/2,2,1/2,1,1,1/4,  8,1,2,4,1,3,4,1),
                 nrow=8, byrow=TRUE)
cons2 <- consEval(pcm2)
cons2
pcm3 <- createLogicalPCM(7)
cons3 <- consEval(pcm3)
print(paste(formatC(cons3$logitConsistency,format="e",digits=4),
         formatC(cons3$prop3Rev,format="f",digits=4),
         formatC(cons3$max3Rev,format="f", digits=4)))
```

`CR` *Saaty CR Consistency*

Description

Computes and returns the Consistency Ratio for an input PCM and its boolean status of consistency based on Consistency Ratio

Usage

```
CR(PCM, typePCM = TRUE)
```

Arguments

`PCM`	A pairwise comparison matrix
`typePCM`	boolean flag indicating if the first argument is a PCM or a vector of upper triangular elements

Value

A list of 3 elements, a boolean for the 'CR' consistency of the input 'PCM', the 'CR' consistency value and the principal eigenvector

Examples

```
CR.pcm1 <- CR(matrix(
                 c(1,1,7,1,1, 1,1,5,1,1/3, 1/7,1/5,1,1/7,1/8, 1,1,7,1,1,
                 1,3,8,1,1), nrow=5, byrow=TRUE))
CR.pcm1
CR.pcm1a <- CR(c(1,7,1,1, 5,1,1/3, 1/7,1/8, 1), typePCM=FALSE)
CR.pcm1a
```

```
CR.pcm2 <- CR(matrix(
                  c(1,1/4,1/4,7,1/5, 4,1,1,9,1/4, 4,1,1,8,1/4,
                  1/7,1/9,1/8,1,1/9, 5,4,4,9,1), nrow=5, byrow=TRUE))
CR.pcm2
CR.pcm2a <- CR(c(1/4,1/4,7,1/5, 1,9,1/4, 8,1/4, 1/9),typePCM=FALSE)
CR.pcm2a
```

`createLogicalPCM` *Simulated Logical Pairwise Comparison Matrix for the Analytic Hierarchy Process*

Description

Creates a logical pairwise comparison matrix for the Analytic Hierarchy Process such as would be created by a rational decision maker based on a relative vector of preferences for the alternatives involved. Choices of the pairwise comparison ratios are from the Fundamental Scale and simulate a reasonable degree of error. The algorithm is modified from a paper by Bose, A [2022], doi:10.1002/mcda.1784

Usage

```
createLogicalPCM(ord, prefVec = rep(NA, ord), granularityLow = TRUE)
```

Arguments

`ord`	The desired order of the Pairwise Comparison Matrix
`prefVec`	The preference vector of length as the order of the input matrix
`granularityLow`	The Scale for pairwise comparisons; default (TRUE) is the fundamental scale; else uses a more find grained scale, derived from pairwise ratios of the elements of the Fundamental Scale.

Value

A Logical Pairwise Comparison Matrix

Examples

```
lPCM <- createLogicalPCM(3,c(1,2,3));
lPCM <- createLogicalPCM(5,c(0.25,0.4,0.1,0.05,0.2));
```

---

createPCM *Create a Pairwise Comparison Matrix of order n for Analytic Hierarchy Process from a vector of length n(n-1)/2 comparison ratios*

Description

Create a Pairwise Comparison Matrix of order n from a vector of length n(n-1)/2 independent upper triangular elements

Usage

```
createPCM(vec)
```

Arguments

`vec`	The preference vector of length as the order of the 'PCM'

Value

A Pairwise Comparison Matrix corresponding to the upper triangular elements

Examples

```
PCM <- createPCM(c(1,2,0.5,3,0.5,2));
PCM <- createPCM(c(1,.5,2,1/3,4,2,.25,1/3,.5,1,.2,6,2,3,1/3));
```

---

improveCR *Improve the CR consistency of a PCM*

Description

For an input pairwise comparison matrix, PCM that is inconsistent, this function returns a consistent PCM if possible, with the relative preference for its alternatives as close as possible to the original preferences, as in the principal right eigenvector.

Usage

```
improveCR(PCM, typePCM = TRUE)
```

Arguments

`PCM`	A pairwise comparison matrix
`typePCM`	boolean flag indicating if the first argument is a PCM or a vector of upper triangular elements

Value

A list of 4 elements, suggested PCM, a boolean for the CR consistency of the input PCM, the CR consistency value, a boolean for the CR consistency of the suggested PCM, the CR consistency value of the suggested PCM

Examples

```
CR.suggest2 <- improveCR(matrix(
                 c(1,1/4,1/4,7,1/5, 4,1,1,9,1/4, 4,1,1,8,1/4,
                 1/7,1/9,1/8,1,1/9, 5,4,4,9,1), nrow=5, byrow=TRUE))
CR.suggest2
CR.suggest2a <- improveCR(c(1/4,1/4,7,1/5, 1,9,1/4, 8,1/4, 1/9),
typePCM=FALSE)
CR.suggest2a
CR.suggest3 <- improveCR(matrix(
                 c(1,7,1,9,8, 1/7,1,1/6,7,9, 1,6,1,9,9, 1/9,1/7,1/9,1,5,
                 1/8,1/9,1/9,1/5,1), nrow=5, byrow=TRUE))
CR.suggest3
```

`logitModel` *Logistic Regression Model*

Description

A logistic regression model object created from the `glm` function.

Usage

```
logitModel
```

Format

An object of class `glm` containing the results of a logistic regression.

Source

Generated from the dataset `myData`.

Examples

```
data(logitModel)
summary(logitModel)
```

`revisedConsistency` *Evaluate Revised Consistency*

Description

This function returns the revised consistency classification for a PCM, evaluated by comparison with the threshold of consistency for intentional PCMs in the same preference heterogeneity quartile. The measure for inconsistency is the geometric mean of ratios in comparison with the corresponding benchmark PCM.

Usage

```
revisedConsistency(PCM, typePCM = TRUE)
```

Arguments

`PCM`	A pairwise comparison matrix
`typePCM`	boolean flag indicating if the first argument is a PCM or a vector of upper triangular elements

Value

A list of four elements, revCons = the revised consistency classification, inconGM = the Geometric Mean measure of inconsistency with the best 'PCM', dQrtl = the preference heterogeneity quartile for the normalized eigenvector, and diff = the preference heterogeneity measure

Examples

```
revCon1 <- revisedConsistency(matrix(
               c(1,1/4,1/4,7,1/5, 4,1,1,9,1/4, 4,1,1,8,1/4,
               1/7,1/9,1/8,1,1/9, 5,4,4,9,1), nrow=5, byrow=TRUE))
revCon1
revCon2 <- revisedConsistency(c(7,1,9,8, 1/6,7,9, 9,9, 5), typePCM=FALSE)
revCon2
```

`sensitivity` *Compute Sensitivity*

Description

This function returns a sensitivity measure for an input pairwise comparison matrix, PCM. Sensitivity is measured by Monte Carlo simulation of 500 PCMs which are perturbations of the input PCM. The perturbation algorithm makes a random choice from one of the 5 closest items in the Fundamental Scale {1/9, 1/8, ... 1/2, 1, 2, ... 8, 9} for each element in the PCM, ensuring the the pairwise reciprocity is maintained. The sensitivity measure is the average Spearman's rank correlation of the vector of ranks of the principal eigenvectors of (i) the input PCM and (ii) the perturbed PCM. The average of the 500 such rank correlations is reported as the measure of sensitivity.

Usage

```
sensitivity(PCM, typePCM = TRUE, granularityLow = TRUE)
```

Arguments

`PCM`	A pairwise comparison matrix
`typePCM`	boolean flag indicating if the first argument is a PCM or a vector of upper triangular elements
`granularityLow`	The Scale for pairwise comparisons; default (TRUE) is the fundamental scale; else uses a more find grained scale, derived from pairwise ratios of the elements of the Fundamental Scale.

Value

The average Spearman's rank correlation between the principal eigenvectors of the input and the perturbed 'PCMs'

Examples

```
revcons1 <- revisedConsistency(matrix(
                 c(1,1/4,1/4,7,1/5, 4,1,1,9,1/4, 4,1,1,8,1/4,
                 1/7,1/9,1/8,1,1/9, 5,4,4,9,1), nrow=5, byrow=TRUE))
revcons1
sensitivity2 <- sensitivity(matrix(
                 c(1,7,1,9,8, 1/7,1,1/6,7,9, 1,6,1,9,9, 1/9,1/7,1/9,1,5,
                 1/8,1/9,1/9,1/5,1), nrow=5, byrow=TRUE))
sensitivity2
```

`viewAHPtree` *Create AHP Tree Structure*

Description

Builds a hierarchical tree from a flat AHP representation.

Usage

```
viewAHPtree(ahp)
```

Arguments

`ahp`	A data frame with the AHP structure including Node and Parent columns

Details

For an overview and examples, please see the associated vignette: 'vignette("viewAHPtree", package = "AHPtools")'

Value

A ‘Node‘ object (from the ‘data.tree‘ package) representing the full AHP tree

Examples

```
file <- system.file("extdata", "example_transport.xlsx", package = "AHPtools")
AHPstruc <- readxl::read_excel(file, sheet = "ahp")
tree <- viewAHPtree(AHPstruc)
print(tree, "level", limit=NULL)
```

Index