Package 'dexisensitivity'

October 30, 2024

Title 'DEXi' Decision Tree Analysis and Visualization

Version 1.0.1

Description
Provides a versatile toolkit for analyzing and visualizing 'DEXi' (Decision EXpert for education) decision trees, facilitating multi-criteria decision analysis directly within R. Users can read .dxi files, manipulate decision trees, and evaluate various scenarios. It supports sensitivity analysis through Monte Carlo simulations, one-at-a-time approaches, and variance-based methods, helping to discern the impact of input variations. Additionally, it includes functionalities for generating sampling plans and an array of visualization options for decision trees and analysis results. A distinctive feature is the synoptic table plot, aiding in the efficient comparison of scenarios. Whether for in-depth decision modeling or sensitivity analysis, this package stands as a comprehensive solution. Definition of sensitivity analyses available in Carpani, Bergez and Monod (2012) <doi:10.1016/j.envsoft.2011.10.002> and detailed description of the package soon available in Alaphilippe, Allart, Carpani, Cavan, Monod and Bergez (submitted to Software Impacts).

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports methods, XML, AlgDesign, grDevices, graphics, plotrix, genalg, dplyr, ggplot2, testthat (>= 3.0.0), xml2

Suggests sloop, vdiffr

Config/testthat/edition 3

LazyData true

NeedsCompilation no

Author Roland Allart [aut],
Jacques-Eric Bergez [aut] (<https://orcid.org/0000-0003-3467-2617>),
Marta Carpani [aut],
Hervé Monod [aut] (<https://orcid.org/0000-0001-8225-495X>),
Aude Alaphilippe [ctb] (<https://orcid.org/0000-0002-4504-2728>),
Nicolas Cavan [ctb, cre],
INRAE [cph] ((National Research Institute for Agriculture, Food and Environment, France))

Maintainer Nicolas Cavan <nicolas.cavan@inrae.fr>

Repository CRAN

Date/Publication 2024-10-29 23:10:02 UTC

Contents

aov_tree *Dual Order AOV on a Decision Tree*

Description

Conducts an Analysis of Variance (AOV) on a provided decision tree, computing both first-order and second-order effects.

Usage

```
aov_tree(tree)
```

Arguments

tree	Tree object to be analyzed.

Value

A list containing results for both first-order and second-order AOV analyses.

Examples

```
tree <- dexisensitivity::masc2
subtree <- create_sub_tree(tree, "Dimension sociale")
AOV_out <- aov_tree(subtree)
```

compare_scenarios *Compare Scenarios Using a Radial Plot*

Description

Visualizes the comparison of node values across multiple scenarios with a radial plot. This representation provides an intuitive view of how different scenarios compare for the selected nodes.

Usage

```
compare_scenarios(tree, scenarios_results, nodes_list)
```

Arguments

tree	A Tree object.
scenarios_results	List of numeric vectors with scenario evaluation results.
nodes_list	List of character node names to be compared in the plot.

Value

No return value, called for side effects

See Also

evaluate_scenarios

Examples

```
tree <- dexisensitivity::masc2
option <- create_options(tree, num_options=3, seed = 42)
scenarios <- evaluate_scenarios(tree, option)
compare_scenarios(tree, scenarios,
 c("Dimension economique", "Dimension sociale", "Dimension environnementale"))
```

---

create_list_synoptique

Create a List of Synoptic Plots Based on Different Options

Description

Produces a list of synoptic plots, one for each set of options specified in the columns of the 'options' matrix. Each plot visualizes nodes of a decision tree based on different evaluation options.

Usage

```
create_list_synoptique(tree, options, depth = NA)
```

Arguments

tree	Tree structure object representing the decision tree.
options	matrix where each column denotes a distinct set of options for evaluating the decision tree.
depth	Optional numeric specifying the depth of the tree. If provided, a sub-tree is produced up to the specified depth. By default, it is set to NA.

Value

A list of ggplot objects. Each item in the list is a synoptic plot corresponding to a column from the 'options' matrix.

See Also

create_synoptique

Examples

```
tree <- dexisensitivity::masc2
options <- create_options(tree, num_options=3, seed = 42)
create_list_synoptique(tree, options)
```

`create_options` *Generate Random Options Matrix for a Given Tree*

Description

Creates random options based on the `RangeScale` and `Probability` attributes of tree nodes. This function is useful for generating random scenarios for simulations or analyses.

Usage

```
create_options(tree, num_options = 1, seed = NULL)
```

Arguments

`tree`	A `Tree` object.
`num_options`	A single `numeric` value specifying the number of options to generate. Defaults to 1.
`seed`	A single `numeric` value for random number generation seed. Default is `NULL`, which means no seed will be set.

Details

The function creates a matrix of random options based on the attributes of tree nodes. Specifically, it leverages the `RangeScale` and `Probability` attributes of tree nodes to generate random options.

Value

A `matrix` where rows represent tree leaves and columns represent sampled options.

Examples

```
tree <- dexisensitivity::masc2
option <- create_options(tree, num_options=3, seed = 42)
```

create_sub_tree *Create a Subtree Starting from a Specific Node*

Description

Constructs a subtree from a specified Node within a given Tree. This subtree encompasses all descendant attributes of the Node.

Usage

```
create_sub_tree(tree, node_name, avoid_repetition = FALSE)
```

Arguments

tree	a Tree object from which the subtree is derived.
node_name	A character indicating the name of the starting Node for the subtree.
avoid_repetition	A logical indicating whether to avoid node repetitions. Default is FALSE.

Value

A new Tree object representing the subtree.

See Also

Relevant functions and classes that provide more context or might be of interest:

- Tree-class: For an in-depth understanding of the Tree class.
- Node-class: To get more details about the structure of a Node.

Examples

```
tree <- dexisensitivity::masc2
subtree <- create_sub_tree(masc2, masc2@Nodes[[2]]@Name)
subtree

# Equivalent to :
tree <- dexisensitivity::masc2
subtree <- create_sub_tree(masc2, "Dimension economique")
subtree
```

create_synoptique *Create a Synoptic Plot from a Given Tree Structure*

Description

Generates a synoptic plot based on a specified tree structure, providing visualization of its structure and evaluation options.

Usage

```
create_synoptique(tree, option, depth = NA)
```

Arguments

tree	Tree structure object.
option	list containing evaluation options for the tree.
depth	numeric representing the depth of the tree; if provided, produces a sub-tree.

Value

A ggplot object representing the synoptic plot.

Examples

```
tree <- dexisensitivity::masc2
option <- create_options(tree, num_options=1, seed = 42)
create_synoptique(tree, option)
```

create_tree *Tree Construction from DEXi's XML Output*

Description

Constructs a Tree objects using the XML representation from a DEXi model.

Usage

```
create_tree(main_tree, which_root = "all", correct = FALSE)
```

Arguments

main_tree	A character or an object of class XML. Either the path to the .dxi file containing the tree (character) or the XML object that represents the primary tree structure and should encompass the DEXi decision model.
which_root	an integer to indicate which tree to create among several in a .dxi or "all" (default) to create them all.
correct	A logical. If TRUE, use some correction function to avoid special character as " ' ".

Details

Iterates over every root attribute in the provided XML to create tree structures that depict the hierarchical decision model.

Value

A list of Tree objects, each corresponding to a unique root attribute in the DEXi XML layout. Each tree details the root attribute, nodes, leaves, depth, and other associated data regarding the decision model's configuration.

Examples

```
# With a Path
dxi_masc2_path <- system.file("extdata", "arborescence_MASC_2_0.dxi", package = "dexisensitivity")
tree <- create_tree(dxi_masc2_path)
tree

# With an XML
dxi_masc2_path <- system.file("extdata", "arborescence_MASC_2_0.dxi", package = "dexisensitivity")
masc2_xml <- XML::xmlDeserializeHook(dxi_masc2_path)
tree <- create_tree(masc2_xml)
tree
```

---

describe *Generic Describe Function for Objects*

Description

Provides a comprehensive description of an object.

Usage

```
describe(object)
```

Arguments

object	The object to be described.

Value

No return value, called for the side effect of a description of the object (description of each node in case of a Tree object)

See Also

- Tree-class: For the Tree class definition.
- print.Tree: For printing a Tree object.
- show.Tree: For showing a Tree object.

---

describe,Tree-method *Describe Method for Tree Class Objects*

Description

Outputs a detailed structure of a Tree object, with each node and its properties displayed distinctly.

Usage

```
## S4 method for signature 'Tree'
describe(object)
```

Arguments

object	The Tree object to be described.

Details

Each node within the Tree is presented separately. If the Tree lacks nodes, an error "Tree without any node!" is raised.

Value

This function is primarily executed for its side effect of presenting nodes from the Tree object and does not provide a meaningful return value.

See Also

- Tree-class: For the Tree class definition.
- print.Tree: For printing a Tree object.
- show.Tree: For showing a Tree object.

dexifruits_v1 *DEXiFruits_v1 model*

Description

A Tree class object created by dexisensitivity::create_tree function, using the Multi-Attribute Assessment model of the sustainability of cropping systems in arboriculture (DEXiFruits) in its first and generic version.

Usage

```
dexifruits_v1
```

Format

An object of class `Tree` of length 1.

Source

https://means.inrae.fr/outils-emc/dexifruits/telecharger-dexifruits

See Also

Tree-class

estimate_aov_time *Execution Time Estimation for Factorial Simulations*

Description

Estimates the expected execution time for a given number of factorial simulations using the duration taken to run a sample subset of simulations.

Usage

```
estimate_aov_time(tree, test_runs = 50)
```

Arguments

`tree`	`Tree` object upon which the simulations are performed.
`test_runs`	`numeric` denoting the number of simulations utilized for estimating the execution time. Default value is set to `50`.

Value

A `character` string, with estimated execution time (in minutes)

Examples

```
tree <- dexisensitivity::masc2
subtree <- create_sub_tree(tree, "Dimension sociale")
estimate_aov_time(subtree, test_runs = 50)
```

---

estimate_mc_time *Estimates Execution Time for Monte Carlo Simulations*

Description

Estimates the time required to run a set number of Monte Carlo simulations based on the time taken to run a smaller test set.

Usage

```
estimate_mc_time(tree, num_runs, num_test = 50)
```

Arguments

tree	A Tree object for simulations.
num_runs	A Numeric, total number of simulations desired.
num_test	A Numeric, number of test simulations for time estimation (default is 50).

Value

A character string with estimated execution time (in minutes)

Examples

```
tree <- dexisensitivity::masc2
estimate_mc_time(tree, num_runs = 1000, num_test = 50)
```

---

evaluate_scenario *Evaluate Node Values in a Tree*

Description

Calculates the values of the tree nodes using a bottom-up approach. The function starts by assigning values to the leaves, based on the provided option, and then aggregates these values up the tree to determine each node's value. This ensures each node's value considers the values of its child nodes.

Usage

```
evaluate_scenario(tree, option)
```

Arguments

tree	A Tree object.
option	A matrix representation of a scenario, providing values for the tree's leaves. Each column in the matrix corresponds to a tree leaf, and the rows provide different values for the scenario analysis.

Details

Begins by assigning values to the leaves of the tree based on the provided option. If the tree structure indicates leaf-aggregated scenarios, these values are aggregated accordingly. The function then continues to aggregate values up the tree, considering the tree's structure, to determine each node's value.

It's essential for the input option matrix to have columns that correspond to the leaves of the tree and for the tree object to have the appropriate attributes set.

Value

A named numeric vector representing the evaluated values for all nodes, progressing from the leaves to the root.

Examples

```
tree <- dexisensitivity::masc2
option <- create_options(tree, num_options=1, seed = 42)
scenario <- evaluate_scenario(tree, option)
scenario
```

---

evaluate_scenarios *Evaluate Multiple Scenarios for a Given Tree*

Description

Evaluates multiple scenarios simultaneously using the evaluate_scenario function. Each scenario is represented as a column in the options_matrix.

Usage

```
evaluate_scenarios(tree, options_matrix)
```

Arguments

tree	A Tree object for evaluation.
options_matrix	A matrix where each column represents a scenario.

Value

A list of numeric vectors with evaluation results for each scenario.

Examples

```
tree <- dexisensitivity::masc2
option <- create_options(tree, num_options=3, seed = 42)
scenarios <- evaluate_scenarios(tree, option)
scenarios
```

`get_sensitivity_index` *Compute Sensitivity Index for Decision Tree*

Description

Computes the Sensitivity Index (SI) for each attribute within the specified tree.

Usage

```
get_sensitivity_index(tree, avoid_repetition = FALSE)
```

Arguments

`tree`	A `Tree` object on which the analysis is to be performed.
`avoid_repetition`	`logical` indicating if repeated nodes should be disregarded. Defaults to FALSE.

Value

A `vector` containing the Sensitivity Indices for each attribute in the tree.

`load_options` *Load Options Table from a File*

Description

Retrieves a matrix of options saved in a file. This matrix can then be used for further analysis or processing.

Usage

```
load_options(file_name)
```

Arguments

`file_name`	A `character` string specifying the file from which to load the options matrix.

Value

A `matrix` representing the loaded options.

Examples

```
tree <- dexisensitivity::masc2
option <- create_options(tree, num_options=3, seed = 42)
save_options(option,paste0(tempdir(),"\\save_options.tab"))
loaded_option <- load_options(paste0(tempdir(),"\\save_options.tab"))
file.remove(paste0(tempdir(),"\\save_options.tab"))
```

---

masc2 *MASC 2.0 model*

Description

A Tree class object created by dexisensitivity::create_tree function, using the Multi-Attribute Assessment of the Sustainability of Cropping systems (MASC) model in it's 2.0 version.

Usage

```
masc2
```

Format

An object of class `Tree` of length 1.

Source

https://means.inrae.fr/outils-emc/masc/telecharger-masc

See Also

Tree-class

---

monte_carlo *Monte Carlo Simulation on a Decision Tree*

Description

Conducts a Monte Carlo simulation over a provided decision tree for a specified number of runs. Optionally, the function can save the random options selected for the analysis in a .csv file named "MC options.csv".

Usage

```
monte_carlo(tree, num_runs, write_to_file = NULL)
```

Arguments

`tree`	`Tree` object to be used in the simulation.
`num_runs`	`numeric` indicating the number of Monte Carlo simulations to be executed.
`write_to_file`	`character` Name of the file created to save the Monte Carlo's results. If `NULL`, don't write any file. Default is `NULL`

Value

A `matrix` containing the results of the Monte Carlo simulation.

Examples

```
tree <- dexisensitivity::masc2
MC <- monte_carlo(tree, 100)
```

`Node-class` *Node Class Definition*

Description

An S4 class to represent a node in a tree structure.

Details

A structured representation of a node, which encompasses various attributes such as name, depth, and relationships with other nodes in the tree structure. The class `Node` is primarily used in the creation, manipulation, and display of nodes within tree structures.

Value

An object of class `Node`.

Slots

`Id numeric` - Unique sequential identifier for the node.

`Name character` - Name of the node.

`IsLeaf logical` - Flag indicating if the node is a leaf.

`IsLeafAndAggregated logical` - Flag indicating if the node is both a leaf and an aggregated node.

`Children character` - List of the node's children names.

`Sisters character` - List of the node's sisters names.

`Mother character` - Name of the node's mother.

`Aggregation matrix` - Aggregation table if the node is aggregated.

`Probability numeric` - Estimated weight for aggregation.

`Depth numeric` - Depth of the node in the tree.

`Twin numeric` - ID of the other leaves for nodes with multiple leaves.

`ConditionalProbabilityList list` - List storing conditional probabilities.

`RangeScale numeric` - Range scale for the node.

`ScaleLabel character` - Labels corresponding to different scales.

`NodePath character` - Path from the root to the leaf for the node.

See Also

- `print.Node`: For printing a Node object.

`oat` *OFAT Sensitivity Analysis*

Description

Executes a One-Factor-At-A-Time (OFAT) sensitivity assessment by undertaking simulations while varying individual factors. During each simulation, all factors are maintained constant save for one.

Usage

```
oat(tree, option)
```

Arguments

`tree`	`Tree` object designated for analysis.
`option`	Initial configuration for `Tree` parameters.

Value

A `matrix` depicting the assessment outcomes for every attribute in the `Tree`, corresponding to diverse parameter alterations.

Examples

```
tree <- dexisensitivity::masc2
optionOAT <- create_options(tree, 1)
results <- oat(tree, optionOAT)
```

plot_sensitivity_index *Show Sensitivity Index (SI)*

Description

Visualizes the Sensitivity Index (SI) of the leaves of a specified decision tree using a bar plot.

Usage

```
plot_sensitivity_index(tree, sensitivity_indices)
```

Arguments

`tree`	`Tree` object for analysis.
`sensitivity_indices`	A `numeric` vector representing the Sensitivity Indices for each leaf in the decision tree.

Value

No return value; a bar plot is displayed.

Examples

```
tree <- dexisensitivity::masc2
sensitivity_index <- si_dexi(tree)
plot_sensitivity_index(tree, sensitivity_index[[1]])
```

print,Node-method *Print Method for Node Class Object*

Description

Provides a comprehensive display of a Node's properties, such as its name, ID, depth, path, and more. This method is intended for better readability and understanding of a Node's structure and relationships.

Usage

```
## S4 method for signature 'Node'
print(x, ...)
```

Arguments

x	An object of class Node that you want to print.
...	Additional arguments to be passed to the underlying print function, though they might not have any effect in this custom print method.

Value

This function is invoked for its side effect of printing. It does not return anything.

See Also

- Node-class: For more details on the Node class.

---

print,Tree-method *Print Method for Tree Class Objects*

Description

Custom print method for objects of class Tree. This method prints out basic information about the tree, such as the root name, number of attributes, number of leaves, depth, and details about specific nodes.

Usage

```
## S4 method for signature 'Tree'
print(x, ...)
```

Arguments

x	An object of class Tree that you want to print.
...	Additional arguments to be passed to the underlying print function, though they might not have any effect in this custom print method.

Value

This function is invoked for its side effect of printing. It does not return anything.

See Also

- Tree-class: For the Tree class definition.
- show.Tree: For showing a Tree object.
- describe.Tree: For describing a Tree object.

save_options *Save Options Table*

Description

Stores a matrix of options into a file, primarily for archival or subsequent analysis.

Usage

```
save_options(options_table, file_name)
```

Arguments

options_table	A matrix containing option values.
file_name	A character string specifying the desired file name for saving the options.

Value

No return value, called for side effects

Examples

```
tree <- dexisensitivity::masc2
option <- create_options(tree, num_options=3, seed = 42)
save_options(option,paste0(tempdir(),"\\save_options.tab"))
file.remove(paste0(tempdir(),"\\save_options.tab"))
```

save_scenarios *Save Evaluation Results of Scenarios to a File*

Description

Stores the results of scenario evaluations into a file for later analysis.

Usage

```
save_scenarios(scenarios_results, file_name)
```

Arguments

scenarios_results	List of numeric vectors with scenario evaluation results.
file_name	A character, to name the file for saving the scenario results.

Value

No return value, called for side effects

Examples

```
tree <- dexisensitivity::masc2
option <- create_options(tree, num_options=3, seed = 42)
scenarios <- evaluate_scenarios(tree, option)
save_scenarios(scenarios,paste0(tempdir(),"\\save_scenarios.tab"))
file.remove(paste0(tempdir(),"\\save_scenarios.tab"))
```

---

show,Tree-method *Show Method for Tree Class Objects*

Description

Custom show method for objects of class `Tree`. It presents a structured representation of the tree using a specific format.

Usage

```
## S4 method for signature 'Tree'
show(object)
```

Arguments

`object`	An object of class `Tree` to be shown.

Details

Each node of the tree is displayed with its depth, name, and associated twin attributes. Node presentation differs based on its type and position within the tree structure:

- Prefix "Z : " is used for the first node of the tree.
- Prefix "X : " denotes a leaf node.
- Prefix "Y : " indicates a non-leaf node.

If a tree has no attributes, it displays "*** Tree without attributes ***".

Value

This function is invoked for its side effect of showing a structured display of the tree. It does not return anything explicitly.

See Also

- `print.Tree`: For printing a Tree object.
- `Tree-class`: For the Tree class definition.

show_mc_results *Visualization of Monte Carlo Simulation Results*

Description

Displays the outcomes of the Monte Carlo simulation for a specific Node as a bar chart, showcasing the frequency of each result. Furthermore, the lengths of the bars are saved in a .csv file titled "MC bar lengths.csv".

Usage

```
show_mc_results(node, mc_results, num_runs, save = NULL)
```

Arguments

node	Node object representing the node of interest in the simulation.
mc_results	matrix containing the Monte Carlo simulation results.
num_runs	numeric indicating the number of Monte Carlo simulations that were executed.
save	character indicating where to save the graphic. By default, save is NULL and don't save the graphic.

Value

A vector depicting the data used in the bar chart.

Examples

```
tree <- dexisensitivity::masc2
MC <- monte_carlo(tree, 100)
show_mc_results(tree@Nodes[[2]], MC, 100)
```

show_oat_results *Visualize OFAT Sensitivity Analysis Outcomes*

Description

Renders a visualization representing the One-Factor-At-A-Time (OFAT) sensitivity analysis results.

Usage

```
show_oat_results(node_name, results, tree)
```

Arguments

node_name	character specifying the name of the node intended for visualization.
results	Matrix of evaluation results, typically derived from ofat_sensitivity_analysis.
tree	A Tree object that was employed in the OFAT analysis.

Value

This function does not return a value; instead, it exhibits a plot.

Examples

```
tree <- dexisensitivity::masc2
optionOAT <- create_options(tree, 1)
results <- oat(tree, optionOAT)
show_oat_results("Dimension economique", results, tree)
```

---

show_scenario *Plot a Bar Chart for a Single Scenario*

Description

Visualizes the attribute values of a provided scenario. For each attribute, a bar is plotted, and the maximum possible value is highlighted.

Usage

```
show_scenario(scenario, tree, label_y = TRUE, modify_par = TRUE)
```

Arguments

scenario	Scenario data to visualize.
tree	Associated Tree object providing attribute details.
label_y	Logical value indicating whether to label the Y-axis (default is TRUE).
modify_par	Logical value to decide if graphical parameters should be modified (default is TRUE).

Value

No return value, called for side effects

See Also

evaluate_scenario

Examples

```
tree <- dexisensitivity::masc2
option <- create_options(tree, num_options=1, seed = 42)
scenario <- evaluate_scenario(tree, option)
show_scenario(as.matrix(scenario), tree = tree, label_y = TRUE)
```

`si_dexi` *Compute Sensitivity Index (SI) for Decision Tree*

Description

Computes the Sensitivity Index (SI) for each node within a specified decision tree.

Usage

```
si_dexi(
  tree,
  file_name = "SI_out.csv",
  is_file = FALSE,
  avoid_repetition = FALSE
)
```

Arguments

`tree`	A `Tree` object on which the analysis will be executed.
`file_name`	`character` designating the filename to store the SI. Default is "SI_out.csv".
`is_file`	`logical` determining whether to save the SI results to a file. Defaults to FALSE
`avoid_repetition`	`logical` indicating if repeated nodes should be disregarded. Defaults to FALSE.

Value

A list containing the Sensitivity Indices for every node in the tree.

Examples

```
tree <- dexisensitivity::masc2
sensitivity_index <- si_dexi(tree)
sensitivity_index
```

`Tree-class` *Tree Class Definition*

Description

An S4 class to represent a tree structure.

Details

A structured representation of a tree, which includes several slots to store attributes, leaves, nodes, path, and additional information related to the tree structure. The class `Tree` is primarily used in the creation, manipulation, and display of tree structures.

Value

An object of class `Tree`.

Slots

`NumberOfAttributes numeric` - Number of attributes in the tree.

`NumberOfLeaves numeric` - Number of leaves in the tree.

`Depth numeric` - Maximum depth of the tree.

`Attributes character` - Names of attributes in the tree.

`Leaves character` - Names of leaves in the tree.

`Aggregated character` - Names of aggregated nodes in the tree.

`IsMultiple logical` - Flag indicating if multiple leaves are present in the tree.

`Multiple data.frame` - List of multiple leaves and their count.

`IsLeafAggregated logical` - Flag indicating if leaf-aggregated nodes are present in the tree.

`LeafAggregated character` - Names of leaf-aggregated nodes in the tree.

`Paths list` - Paths from the root to the leaves.

`Nodes list` - Nodes present in the tree.

`EvaluationOrder numeric` - Evaluation order for LeafAggregated nodes.

`RootName character` - Name of the root node.

See Also

- `print.Tree`: For printing a Tree object.
- `show.Tree`: For showing a Tree object.
- `describe.Tree`: For describing a Tree object.

visualize_aov *Visualize AOV Outcomes*

Description

Renders the outcomes of an Analysis of Variance (AOV) through bar plots, allowing a comprehensive display of both total sums and specific effects.

Usage

```
visualize_aov(
  aov_results,
  show_main = TRUE,
  num_plots = 8,
  horizontal = TRUE,
  axis_label_style = 1,
  ...
)
```

Arguments

aov_results	A list containing the AOV results.
show_main	logical indicating if the main effects and total sums of squares should be included in the visualization. Default is TRUE.
num_plots	numeric specifying the count of effects to be showcased in the display.
horizontal	logical determining if the bar plots should be oriented horizontally. Default is TRUE.
axis_label_style	numeric designating the label styling for the plot's axes. Default is 1.
...	Additional arguments affecting the bar plot's aesthetics.

Value

A data.frame containing proportions derived from the sum of squares.

Examples

```
tree <- dexisensitivity::masc2
subtree <- create_sub_tree(tree, "Dimension sociale")
AOV_out <- aov_tree(subtree)
visualize_aov(AOV_out)
```

Index