# Package ‘topologyR’

October 3, 2025

**Type** Package

**Title** Topological Connectivity Analysis for Numeric Data

**Version** 0.1.1

**Description** Description: Implementation of topological data analysis methods based on graph-theoretic approaches for discovering topological structures in data. The core algorithm constructs topological spaces from graphs following Nada et al. (2018) <doi:10.1002/mma.5096> ``New types of topological structures via graphs".

**License** MIT + file LICENSE

**URL** `https://github.com/IsadoreNabi/topologyR`

**BugReports** `https://github.com/IsadoreNabi/topologyR/issues`

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Depends** R (>= 4.0.0)

**Imports** ggplot2, stats, utils

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** José Mauricio Gómez Julián [aut, cre]

**Maintainer** José Mauricio Gómez Julián <`isadore.nabi@pm.me`>

**Repository** CRAN

**Date/Publication** 2025-10-03 12:20:02 UTC

## Contents

---

## `analyze_topology_factors` *Calculate topology characteristics for different IQR factors*

### Description

This function analyzes how different IQR (Interquartile Range) factors affect the topology's characteristics. It helps users determine the optimal factor for their specific data by showing how the factor choice impacts the base size and set sizes in the resulting topology.

### Usage

```
analyze_topology_factors(data, factors = NULL, plot = TRUE)
```

### Arguments

| | |
|---|---|
| `data` | Numeric vector containing the data to analyze |
| `factors` | Numeric vector of factors to test (default: c(1, 2, 4, 8, 16)) |
| `plot` | Logical, whether to display a plot (default: TRUE) |

### Details

The function works by:

1. Calculating different thresholds using IQR/factor
2. Creating a subbase using these thresholds
3. Generating the base from intersections of subbase elements
4. Analyzing the resulting topology's characteristics

A larger factor results in a smaller threshold, which typically leads to a finer topology with more distinct sets but smaller set sizes.

### Value

A `data.frame` with the following columns:

**factor** Numeric. The IQR factor used for threshold calculation.

**threshold** Numeric. The calculated threshold value (IQR/factor).

**base_size** Integer. Number of sets in the topological base.

**max_set_size** Integer. Size of the largest set in the base.

**min_set_size** Integer. Size of the smallest set in the base.

If `plot = TRUE`, also generates a line plot showing the relationship between IQR factors and base sizes as a side effect.

## Examples

```
# Generate sample data
data <- rnorm(100)

# Analyze topology with default factors
results <- analyze_topology_factors(data)
print(results)

# Use custom factors
custom_results <- analyze_topology_factors(data, factors = c(2, 4, 6))
print(custom_results)
```

---

# `calculate_thresholds` *Calculate multiple threshold methods for topology analysis*

## Description

Calculate multiple threshold methods for topology analysis

## Usage

```
calculate_thresholds(data)
```

## Arguments

| | |
|---|---|
| `data` | Numeric vector to calculate thresholds for |

## Value

A named `list` containing five threshold values:

**mean_diff** Numeric. Threshold based on mean of absolute differences between adjacent sorted values.

**median_diff** Numeric. Threshold based on median of absolute differences between adjacent sorted values.

**sd** Numeric. Threshold based on standard deviation of the data.

**iqr** Numeric. Threshold based on IQR divided by 4.

**dbscan** Numeric. Threshold based on DBSCAN-like density estimation using k-th nearest neighbor distance.

---

## `calculate_topology` *Calculate topology base size for a given threshold*

---

### Description

Calculate topology base size for a given threshold

### Usage

```
calculate_topology(data, threshold)
```

### Arguments

| | |
|---|---|
| `data` | Numeric vector to analyze |
| `threshold` | Numeric value for the threshold parameter |

### Value

`integer` scalar. The number of sets in the topological base. This value represents the complexity of the topology - larger values indicate more complex topological structure.

---

## `complete_topology` *Create a complete topology from data points with neighborhood structure*

---

### Description

Create a complete topology from data points with neighborhood structure

### Usage

```
complete_topology(datos)
```

### Arguments

| | |
|---|---|
| `datos` | Numeric vector containing the data points to analyze |

### Value

A `list` with four components:

**R** List. The equivalence relation defined on the data, where each element represents a relationship between vertices based on their degrees and values.

**subbase** List. The neighborhoods of each vertex, where each element contains the indices of neighboring vertices in the graph.

**base** List. The base generated from intersections of neighborhoods, including the empty set, full set, and all non-empty intersections.

**topology** List. The complete topology generated by taking unions of base elements, satisfying the axioms of topological spaces.

### Examples

```
data <- c(1, 2, 3, 4, 5)
result <- complete_topology(data)
```

---

## `is_topology_connected` *Topology Analysis Functions for Time Series Data*

---

### Description

Functions for analyzing topological properties of time series data

### Usage

```
is_topology_connected(topology)
```

### Arguments

| | |
|---|---|
| `topology` | A list of sets representing the topology |

### Details

This module provides three main functions for analyzing the connectivity of topological structures, particularly focused on economic time series:

- is_topology_connected: Uses an undirected graph approach
- is_topology_connected2: Uses a directed graph approach
- is_topology_connected_manual: Uses a manual checking approach

Created as part of research on economic cycle analysis. Check if a topology is connected using undirected graph approach

### Value

`logical` scalar. Returns `TRUE` if the topology is connected (all elements can be reached from any starting point through the undirected graph representation), `FALSE` otherwise. Returns `FALSE` if the topology is empty.

### Author(s)

José Mauricio Gómez Julián

### Examples

```
topology <- list(c(1,2,3), c(3,4,5))
is_topology_connected(topology)
```

---

## `is_topology_connected2` *Check if a topology is connected using directed graph approach*

---

### Description

Check if a topology is connected using directed graph approach

### Usage

```
is_topology_connected2(topology)
```

### Arguments

| | |
|---|---|
| `topology` | A list of sets representing the topology |

### Value

`logical` scalar. Returns `TRUE` if the topology is connected through directed graph traversal (considering sequential connections between elements), `FALSE` otherwise. Returns `FALSE` if the topology is empty.

### Examples

```
topology <- list(c(1,2,3), c(3,4,5))
is_topology_connected2(topology)
```

---

## `is_topology_connected_manual` *Check if a topology is connected using manual checking approach*

---

### Description

Check if a topology is connected using manual checking approach

### Usage

```
is_topology_connected_manual(topology)
```

### Arguments

| | |
|---|---|
| `topology` | A list of sets representing the topology |

## Value

`logical` scalar. Returns `TRUE` if all elements from 1 to the maximum value in the topology are present in at least one set, `FALSE` otherwise.

## Examples

```
topology <- list(c(1,2,3), c(3,4,5))
is_topology_connected_manual(topology)
```

---

# `simplest_topology` *Create a topology with completely disconnected sets*

## Description

This function generates a topology where each set in the topology is a singleton (contains only one element), resulting in a completely disconnected topological structure. Each vertex exists in isolation, with no meaningful connections between sets.

## Usage

```
simplest_topology(datos)
```

## Arguments

| | |
|---|---|
| `datos` | Numeric vector containing the data points to analyze |

## Details

The subbase contains individual elements. The base consists of singleton sets. The topology is formed by these singleton sets. No meaningful topological relationships are established between elements.

## Value

A `list` with four components:

**R** List. The equivalence relation defined on the data vertices.

**subbase** List. Sets of individual vertices forming the subbase.

**base** List. Singleton sets forming the base of the topology.

**topology** List. The complete topology consisting of disconnected singleton sets.

## Examples

```
data <- c(1, 2, 3, 4, 5)
result <- simplest_topology(data)
```

---

## `visualize_topology_thresholds` *Visualize and Compare Different Threshold Methods*

---

### Description

This function creates a comprehensive visualization of different threshold methods used in topology analysis. It generates three distinct plots to help understand the relationships between different threshold methods and their effects on topological structure.

### Usage

```
visualize_topology_thresholds(data, plot = TRUE)
```

### Arguments

| | |
|---|---|
| `data` | Numeric vector to analyze |
| `plot` | Logical indicating whether to display plots (default: TRUE) |

### Details

Visualize Topology Thresholds

### Value

A `data.frame` with one row per threshold method containing:

**method** Character. Name of the threshold calculation method.

**threshold** Numeric. The calculated threshold value.

**base_size** Integer. Number of sets in the resulting topological base.

If `plot = TRUE`, generates three plots as side effects: (1) Bar chart comparing threshold values by method, (2) Bar chart comparing base sizes by method, (3) Scatter plot showing threshold vs base size relationship.

### Examples

```
# Generate sample data
data <- rnorm(100)

# Visualize threshold comparisons
results <- visualize_topology_thresholds(data)
```

# Index