

Package ‘paneldesc’

May 9, 2026

Type Package

Title Descriptive Analysis and Visualization for Panel Data

Version 0.1.1

Description Provides a comprehensive set of tools for describing and visualizing panel data structures, as well as for summarizing and visualizing variables within a panel data context.

License GPL-3

URL <https://github.com/dtereshch/paneldesc>,
<https://dtereshch.github.io/paneldesc-guides/>

BugReports <https://github.com/dtereshch/paneldesc/issues>

LazyData true

Encoding UTF-8

RoxygenNote 7.3.2

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Dmitrii Tereshchenko [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-8973-542X>>)

Maintainer Dmitrii Tereshchenko <dtereshch@gmail.com>

Repository CRAN

Date/Publication 2026-03-23 17:30:10 UTC

Contents

| | |
|-------------------------------|----|
| decompose_factor | 2 |
| decompose_numeric | 4 |
| describe_balance | 6 |
| describe_dimensions | 8 |
| describe_incomplete | 9 |
| describe_patterns | 11 |

| | |
|--------------------------------|----|
| describe_periods | 13 |
| make_panel | 15 |
| plot_heterogeneity | 16 |
| plot_missing | 18 |
| plot_patterns | 19 |
| plot_periods | 21 |
| production | 22 |
| summarize_missing | 24 |
| summarize_numeric | 25 |
| summarize_transition | 27 |

| | |
|--------------|-----------|
| Index | 30 |
|--------------|-----------|

| | |
|------------------|---|
| decompose_factor | <i>Panel Data Factor Variable Decomposition</i> |
|------------------|---|

Description

This function performs one-way tabulations and decomposes counts into between and within components for categorical (factor) variables in panel data.

Usage

```
decompose_factor(
  data,
  select = NULL,
  index = NULL,
  format = "wide",
  digits = 3
)
```

Arguments

| | |
|--------|--|
| data | A data.frame containing panel data in a long format. |
| select | A character vector specifying which categorical (factor) variables to analyze. If not specified, all factor variables in the data.frame will be used. |
| index | A character vector of length 1 or 2 specifying the names of the entity and (optionally) time variables. The first element is the entity variable; if a second element is provided, it is used as the time variable. Not required if data has panel attributes. |
| format | A character string specifying the output format: "wide" or "long". Default = "wide". |
| digits | An integer indicating the number of decimal places to round shares. Default = 3. |

Details

The output format is controlled by the `format` parameter.

When `format = "wide"` (default), returns a `data.frame` with columns:

`variable` The name of the analyzed variable
`category` The category level of the variable
`count_overall` Overall frequency (person-time observations)
`share_overall` Overall share (`count_overall / total_obs`)
`count_between` Between-entity frequency (number of entities ever having this category)
`share_between` Between-entity share (`count_between / total_entities`)
`share_within` Within-entity share (average share of time entities have this category)

When `format = "long"`, returns a `data.frame` with columns:

`variable` The name of the analyzed variable
`category` The category level of the variable
`dimension` Type of decomposition: "overall", "between", or "within"
`count` Frequency count (NA for within dimension)
`share` Share proportion (0 to 1)

The object has class `"panel_summary"` and two additional attributes:

`metadata` List containing the function name and the arguments used.

`details` List containing additional information: `count_entities`.

Value

A `data.frame` with categorical panel data decomposition statistics.

References

For Stata users: This corresponds to the `xttab` command.

See Also

See also [decompose_numeric\(\)](#), [summarize_transition\(\)](#).

Examples

```
data(production)

# Basic usage
decompose_factor(production, index = "firm")

# With panel_data object
panel <- make_panel(production, index = c("firm", "year"))
decompose_factor(panel)
```

```

# Selecting specific variables
decompose_factor(production, select = "industry", index = "firm")

# Returning results in a long format
decompose_factor(production, index = "firm", format = "long")

# Custom rounding
decompose_factor(production, index = "firm", digits = 2)

# Accessing attributes
out_dec_fac <- decompose_factor(production, index = "firm")
attr(out_dec_fac, "metadata")
attr(out_dec_fac, "details")

```

decompose_numeric *Panel Data Numeric Variable Decomposition*

Description

This function decomposes variance of numeric variables into between and within components in panel data.

Usage

```

decompose_numeric(
  data,
  select = NULL,
  index = NULL,
  detail = TRUE,
  format = "long",
  digits = 3
)

```

Arguments

| | |
|--------|--|
| data | A data.frame containing panel data in a long format. |
| select | A character vector specifying which numeric variables to analyze. If not specified, all numeric variables in the data.frame will be used. |
| index | A character vector of length 1 or 2 specifying the names of the entity and (optionally) time variables. The first element is the entity variable; if a second element is provided, it is used as the time variable. Not required if data has panel attributes. |
| detail | A logical flag indicating whether to return detailed Stata-like output. Default = TRUE. |
| format | A character string specifying the output format: "long" or "wide". Default = "long". |
| digits | An integer indicating the number of decimal places to round statistics. Default = 3. |

Details

The output format is controlled by two parameters: `format` and `detail`.

When `format = "long"` and `detail = TRUE` (default), returns a `data.frame` with:

`variable` The name of the analyzed variable
`dimension` Type of decomposition: "overall", "between", or "within"
`mean` Mean value (only for "overall" row)
`std` Standard deviation
`min` Minimum value
`max` Maximum value
`count` Number of observations or entities

When `format = "long"` and `detail = FALSE`, returns a `data.frame` with:

`variable` The name of the variable
`dimension` Type of decomposition: "overall", "between", or "within"
`mean` Mean value
`std` Standard deviation

When `format = "wide"` and `detail = TRUE`, returns a `data.frame` with:

`variable` The name of the variable
`mean` Overall mean
`std_overall` Overall standard deviation
`min_overall` Overall minimum
`max_overall` Overall maximum
`count_overall` Number of observations
`std_between` Between-entity standard deviation
`min_between` Minimum of entity means
`max_between` Maximum of entity means
`count_between` Number of entities
`std_within` Within-entity standard deviation
`min_within` Within-entity minimum (transformed)
`max_within` Within-entity maximum (transformed)
`count_within` Average observations per entity

When `format = "wide"` and `detail = FALSE`, returns a `data.frame` with:

`variable` The name of the variable
`mean` Overall mean
`std_overall` Overall standard deviation
`std_between` Between-entity standard deviation
`std_within` Within-entity standard deviation

The object has class `"panel_summary"` and two additional attributes:

`metadata` List containing the function name and the arguments used.
`details` List containing additional information: `count_entities`.

Value

A data.frame with panel data decomposition statistics.

References

For Stata users: This corresponds to the xtsum command.

See Also

See also [decompose_factor\(\)](#), [summarize_numeric\(\)](#), [plot_heterogeneity\(\)](#).

Examples

```
data(production)

# Basic usage
decompose_numeric(production, index = "firm")

# With panel_data object
panel <- make_panel(production, index = c("firm", "year"))
decompose_numeric(panel)

# Selecting specific variables
decompose_numeric(production, select = c("sales", "labor"), index = "firm")

# Returning results in a wide format without excessive details
decompose_numeric(production, index = "firm", detail = FALSE, format = "wide")

# Custom rounding
decompose_numeric(production, index = "firm", digits = 2)

# Accessing attributes
out_dec_num <- decompose_numeric(production, index = "firm")
attr(out_dec_num, "metadata")
attr(out_dec_num, "details")
```

describe_balance

Panel Data Balance Description

Description

This function provides summary statistics for panel data structure with focus on balance and data completeness.

Usage

```
describe_balance(data, index = NULL, detail = FALSE, digits = 3)
```

Arguments

| | |
|---------------------|---|
| <code>data</code> | A data.frame containing panel data in a long format. |
| <code>index</code> | A character vector of length 2 specifying the names of the entity and time variables. Not required if data has panel attributes. |
| <code>detail</code> | A logical flag indicating whether to return additional statistics (5th, 25th, 50th, 75th, and 95th percentiles). Default = FALSE. |
| <code>digits</code> | An integer specifying the number of decimal places for rounding mean values. Default = 3. |

Details

The statistics for entities describe the distribution of the number of entities observed per time period (cross-sectional size per period). The statistics for periods describe the distribution of the number of time periods observed per entity (temporal length per entity).

The returned data.frame always contains the following columns:

`dimension` Either "entities" or "periods".

`mean` Mean number of entities per period (or periods per entity).

`std` Standard deviation.

`min` Minimum value.

`max` Maximum value.

When `detail = TRUE`, five additional percentile columns are included:

`p5` 5th percentile.

`p25` 25th percentile (first quartile).

`p50` 50th percentile (median).

`p75` 75th percentile (third quartile).

`p95` 95th percentile.

All statistics are rounded to the number of decimal places specified by `digits`.

The object has class "panel_description" and two additional attributes:

`metadata` List containing the function name and the arguments used.

`details` List containing the full presence matrix.

Value

A data.frame with panel data summary statistics for entities and periods.

Note

An entity-time combination is considered **present** if the corresponding row contains at least one non-NA value in any substantive variable (all columns except the entity and time identifiers).

See Also

See also [describe_dimensions\(\)](#), [describe_periods\(\)](#), [describe_patterns\(\)](#), [plot_periods\(\)](#).

Examples

```
data(production)

# Basic usage
describe_balance(production, index = c("firm", "year"))

# With panel_data object
panel <- make_panel(production, index = c("firm", "year"))
describe_balance(panel)

# Returning detailed statistics
describe_balance(production, index = c("firm", "year"), detail = TRUE)

# Custom rounding
describe_balance(production, index = c("firm", "year"), digits = 2)

# Accessing attributes
out_des_bal <- describe_balance(production, index = c("firm", "year"))
attr(out_des_bal, "metadata")
attr(out_des_bal, "details")
```

describe_dimensions *Panel Data Dimensions Description*

Description

This function provides basic dimension counts for panel data: number of rows, unique entities, unique time periods, and substantive variables.

Usage

```
describe_dimensions(data, index = NULL)
```

Arguments

| | |
|-------|--|
| data | A data.frame containing panel data in a long format. |
| index | A character vector of length 2 specifying the names of the entity and time variables. Not required if data has panel attributes. |

Details

The returned data.frame has the following structure:

rows Total number of rows in the data frame.

entities Number of distinct values in the entity variable.

periods Number of distinct values in the time variable.

variables Number of substantive variables (all columns except entity and time).

The object has class "panel_description" and two additional attributes:

metadata List containing the function name and the arguments used.

details List with the actual vectors of entities, periods, and substantive variables.

Value

A data.frame containing panel dimension counts.

See Also

See also [describe_balance\(\)](#), [describe_periods\(\)](#).

Examples

```
data(production)

# Basic usage
describe_dimensions(production, index = c("firm", "year"))

# With panel_data object
panel <- make_panel(production, index = c("firm", "year"))
describe_dimensions(panel)

# Accessing attributes
out_des_dim <- describe_dimensions(production, index = c("firm", "year"))
attr(out_des_dim, "metadata")
attr(out_des_dim, "details")
```

describe_incomplete *Incomplete Entities Description*

Description

This function provides a descriptive table of entities with incomplete observations (missing values).

Usage

```
describe_incomplete(data, index = NULL, detail = FALSE)
```

Arguments

| | |
|--------|--|
| data | A data.frame containing panel data in a long format. |
| index | A character vector of length 1 or 2 specifying the names of the entity and (optionally) time variables. The first element is the entity variable; if a second element is provided, it is used as the time variable. Not required if data has panel attributes. |
| detail | A logical flag indicating whether to include detailed missing counts for each variable. Default = FALSE. |

Details

The returned data.frame has the following structure:

[entity] The entity identifier (name matches input entity variable)

na_count Total number of missing observations for the entity

variables Number of variables with at least one missing value for that entity

When detail = TRUE, additional columns are included for each substantive variable, showing the number of NAs in that variable for the entity.

The data.frame is sorted by:

1. Number of variables with NAs (descending)
2. Total number of NAs (descending)

The object has class "panel_description" and two additional attributes:

metadata List containing the function name and the arguments used.

details List containing total entity counts and the IDs of incomplete entities.

Value

A data.frame with incomplete entities description.

Note

The interpretation of incomplete entities may differ depending on whether the panel is balanced or unbalanced. In a balanced panel, each entity has the same number of time periods, so the total possible observations per entity are equal. In an unbalanced panel, entities may have different numbers of time periods, so the number of missing values should be interpreted relative to the entity's total observations. The function does not adjust for the number of time periods per entity; the missing counts reflect absolute counts of NAs in the data. Users should consider the panel structure when interpreting the results.

See Also

See also [summarize_missing\(\)](#), [describe_patterns\(\)](#), [describe_periods\(\)](#).

Examples

```

data(production)

# Basic usage with entity only
describe_incomplete(production, index = "firm")

# With time variable (check duplicates)
describe_incomplete(production, index = c("firm", "year"))

# With panel_data object
panel <- make_panel(production, index = c("firm", "year"))
describe_incomplete(panel)

# Returning detailed results
describe_incomplete(production, index = "firm", detail = TRUE)

# Accessing attributes
out_des_inc <- describe_incomplete(production, index = c("firm", "year"))
attr(out_des_inc, "metadata")
attr(out_des_inc, "details")

```

| | |
|-------------------|---|
| describe_patterns | <i>Entities Presence Patterns Description</i> |
|-------------------|---|

Description

This function describes entities presence patterns in panel data over time.

Usage

```

describe_patterns(
  data,
  index = NULL,
  delta = NULL,
  limits = NULL,
  detail = TRUE,
  format = "wide",
  digits = 3
)

```

Arguments

| | |
|-------|--|
| data | A data.frame containing panel data in a long format. |
| index | A character vector of length 2 specifying the names of the entity and time variables. Not required if data has panel attributes. |
| delta | An optional integer giving the expected interval between time periods. |

| | |
|--------|---|
| limits | Either a single integer (show that many most frequent patterns) or a vector of two integers (show patterns with ranks between the two values, inclusive). If not specified, all patterns are shown. |
| detail | A logical flag indicating whether to return detailed patterns. Default = TRUE. |
| format | A character string specifying the output format: "wide" or "long". Default = "wide". |
| digits | An integer specifying the number of decimal places for rounding share column. Default = 3. |

Details

The output format is controlled by `format` and `detail`.

When `format = "wide"` and `detail = TRUE` (default):

`pattern` Pattern number (ranked by frequency).

`[time1]`, `[time2]`, ... Presence (1) / absence (0) for each time period.

`count` Number of entities sharing this pattern.

`share` Proportion of entities with this pattern (rounded to `digits`).

When `format = "wide"` and `detail = FALSE`, only the pattern and presence columns are returned.

When `format = "long"` and `detail = TRUE`:

`pattern` Pattern number.

`[time]` Time period identifier (name equals the original time variable).

`presence` Presence (1) / absence (0).

`count` Number of entities with this pattern.

`share` Proportion of entities with this pattern.

When `format = "long"` and `detail = FALSE`, only pattern, time, and presence columns are returned.

Effect of `delta`: If `delta` is supplied, the function checks that all observed time points are separated by multiples of `delta`. If gaps are detected, a message lists the missing periods (unless the interval was inherited from panel attributes), and columns for those missing periods are added to the presence matrix – and therefore to the output `data.frame` – with all zeros. This ensures that the patterns reflect the full regular sequence of time periods.

The object has class `"panel_description"` and two additional attributes:

`metadata` List containing the function name and the arguments used.

`details` List with the full presence matrix, pattern-entity mapping, and the pattern matrix.

Value

A `data.frame` with presence patterns.

Note

An entity-time combination is considered **present** if the corresponding row contains at least one non-NA value in any substantive variable (i.e., all columns except the entity and time identifiers).

See Also

See also [plot_patterns\(\)](#), [describe_periods\(\)](#), [describe_balance\(\)](#).

Examples

```
data(production)

# Basic usage
describe_patterns(production, index = c("firm", "year"))

# With panel_data object
panel <- make_panel(production, index = c("firm", "year"))
describe_patterns(panel)

# Specifying time interval
describe_patterns(production, index = c("firm", "year"), delta = 1)

# Showing only the top 3 patterns
describe_patterns(production, index = c("firm", "year"), limits = 3)

# Showing patterns ranked 4 to 6
describe_patterns(production, index = c("firm", "year"), limits = c(4, 6))

# Returning results in a long format without excessive details
describe_patterns(production, index = c("firm", "year"), detail = FALSE, format = "long")

# Custom rounding
describe_patterns(production, index = c("firm", "year"), digits = 2)

# Accessing attributes
out_des_pat <- describe_patterns(production, index = c("firm", "year"))
attr(out_des_pat, "metadata")
attr(out_des_pat, "details")
```

describe_periods

Time Periods Completeness Description

Description

This function calculates, for each time period, the number of entities that have at least one non-missing value in any substantive variable, and the corresponding share of all entities.

Usage

```
describe_periods(data, index = NULL, delta = NULL, digits = 3)
```

Arguments

| | |
|---------------------|--|
| <code>data</code> | A <code>data.frame</code> containing panel data in a long format. |
| <code>index</code> | A character vector of length 2 specifying the names of the entity and time variables. Not required if data has panel attributes. |
| <code>delta</code> | An optional integer giving the expected interval between time periods. |
| <code>digits</code> | An integer specifying the number of decimal places for rounding the share column. Default = 3. |

Details

The returned `data.frame` contains the following columns:

`[time]` Time period identifier (name matches the input time variable).

`count` Number of distinct entities observed in that period, i.e., entities with at least one row containing a non-NA value in substantive variables.

`share` Proportion of entities observed in that period (0 to 1), rounded to `digits`.

Effect of `delta`: If `delta` is supplied, the function checks that all observed time points are separated by multiples of `delta`. If gaps are detected, a message lists the missing periods (unless the interval was inherited from panel attributes). For each missing period, a row is added to the output with `count = 0` and `share = 0`, ensuring that the output covers the full regular time sequence.

The object has class `"panel_description"` and two additional attributes:

`metadata` List containing the function name and the arguments used.

`details` List with a named list `entities` giving, for each period, the vector of entities observed.

Value

A `data.frame` with entities presence summary by time period.

See Also

See also [plot_periods\(\)](#), [describe_balance\(\)](#), [describe_patterns\(\)](#).

Examples

```
data(production)

# Basic usage
describe_periods(production, index = c("firm", "year"))

# With panel_data object
panel <- make_panel(production, index = c("firm", "year"))
describe_periods(panel)
```

```

# Specifying time interval
describe_periods(production, index = c("firm", "year"), delta = 1)

# Custom rounding
describe_periods(production, index = c("firm", "year"), digits = 2)

# Accessing attributes
out_des_per <- describe_periods(production, index = c("firm", "year"))
attr(out_des_per, "metadata")
attr(out_des_per, "details")

```

make_panel

Panel Data Structure Setting and Balancing

Description

This function adds panel structure attributes to a data.frame, storing entity and time variable names, and optionally checks the expected interval between time periods. It can also balance the panel with a chosen method.

Usage

```
make_panel(data, index, delta = NULL, balance = NULL)
```

Arguments

| | |
|---------|--|
| data | A data.frame containing panel data in a long format. |
| index | A character vector of length 2 specifying the names of the entity and time variables. |
| delta | An optional integer giving the expected interval between time periods. |
| balance | One of "rows", "entities", or "periods". If specified, the panel is balanced according to the chosen method. |

Details

This function adds attributes to a data.frame to mark it as panel data. The returned object has class "panel_data" and includes the following attributes:

metadata List containing the function name and the arguments used (entity, time, delta, and balance if provided).

details List with diagnostic vectors:

entities Unique values of the entity variable.

periods Sorted unique values of the time variable.

periods_restored, periods_missing If delta is supplied and gaps are detected, the full sequence and missing periods.

Effect of delta: If delta is supplied, the function checks that all observed time points are separated by multiples of delta. If gaps are detected, a message lists the missing periods and the full sequence is stored in `details$periods_restored`.

Balancing the panel (presence definition as in `describe_patterns`):

`balance = "rows"` Create a row for every entity-time combination. If delta is supplied, the full time sequence (including missing periods) is used. Missing combinations get NA in all other columns.

`balance = "entities"` Keep only entities present in **all** time periods.

`balance = "periods"` Keep only time periods where **all** entities are present.

Value

The input `data.frame` with additional attributes, after possibly filtering or expanding rows.

See Also

See also [describe_dimensions\(\)](#), [describe_balance\(\)](#), [describe_periods\(\)](#).

Examples

```
data(production)

# Basic usage
panel <- make_panel(production, index = c("firm", "year"))

# Specifying time interval
panel <- make_panel(production, index = c("firm", "year"), delta = 1)

# Creating balanced panels
panel_bal_ent <- make_panel(production, index = c("firm", "year"), balance = "entities")
panel_bal_per <- make_panel(production, index = c("firm", "year"), balance = "periods")
panel_bal_row <- make_panel(production, index = c("firm", "year"), balance = "rows", delta = 1)

# Accessing attributes
attr(panel, "metadata")
attr(panel, "details")
```

plot_heterogeneity *Heterogeneity Visualization*

Description

This function creates visualizations of heterogeneity among groups.

Usage

```
plot_heterogeneity(data, select, group = NULL, colors = c("darkblue", "gray"))
```

Arguments

| | |
|--------|--|
| data | A data.frame containing variables for analysis. |
| select | A character string specifying the numeric variable of interest. |
| group | A character string or vector of character strings specifying the grouping variable(s). If data has panel attributes and group is not specified, both the entity and time variables will be used as grouping variables. |
| colors | A character vector of two colors: first for mean line and points, second for individual points. Default = c("darkblue", "gray"). |

Details

This function creates one or more plots (depending on the number of grouping variables) showing the heterogeneity among groups. Each plot displays individual observations (points) and group means (connected line).

The returned list contains the following components:

metadata List containing the function name, selection, group, and colors.

details List containing group-level statistics for each grouping variable, each containing means, standard deviations, and counts per group.

Value

Invisibly returns a list with summary statistics and metadata.

See Also

See also [decompose_numeric\(\)](#), [summarize_numeric\(\)](#).

Examples

```
data(production)

# Basic usage with regular data.frame
plot_heterogeneity(production, select = "labor", group = "year")

# Using multiple grouping variables
plot_heterogeneity(production, select = "sales", group = c("firm", "industry", "year"))

# With panel_data object (uses both entity and time)
panel <- make_panel(production, index = c("firm", "year"))
plot_heterogeneity(panel, select = "labor")

# Custom colors
plot_heterogeneity(production, select = "sales", group = "year",
  colors = c("black", "gray"))

# Accessing list components
out_plo_het <- plot_heterogeneity(panel, select = "capital", group = "year")
out_plo_het$metadata
```

```
out_plo_het$details
```

plot_missing *Missing Values Heatmap by Period*

Description

This function creates a heatmap showing the number of missing values for each variable across all time periods in panel data.

Usage

```
plot_missing(data, select = NULL, index = NULL, colors = c("darkblue", "gray"))
```

Arguments

| | |
|--------|---|
| data | A data.frame containing panel data in a long format. |
| select | A character vector specifying which variables to include. If not specified, all substantive variables (except entity and time) are used. |
| index | A character vector of length 2 giving the names of the entity and time variables. Not required if data has panel attributes. |
| colors | A character vector of two colors defining the gradient for the heatmap. The first color represents the largest number of missing values, the second color the smallest number. Default = c("darkblue", "gray"). |

Details

The function creates a heatmap where rows are variables and columns are time periods. Cell color reflects the number of missing values in that variable for that period, using a continuous gradient from colors[1] (most missing) to colors[2] (least missing). Rows are ordered as the variables appear (first at the top). Columns are ordered chronologically.

The returned list contains:

metadata List containing the function call, select, entity/time variables, and colors.

details List with the missing count matrix (variables × periods).

Value

Invisibly returns a list with summary statistics and metadata.

Note

The interpretation of missing counts may differ depending on whether the panel is balanced or unbalanced. In a balanced panel, each time period contains the same number of entities, so the raw NA counts per period are directly comparable across periods. In an unbalanced panel, the number of entities varies by period, so the raw NA counts should be interpreted relative to the number of observations available in each period. The function does not standardize the counts by period size; users should account for the panel structure when interpreting the results.

See Also

See also [summarize_missing](#), [plot_patterns\(\)](#), [plot_periods\(\)](#).

Examples

```
data(production)

# Basic usage
plot_missing(production, index = c("firm", "year"))

# With panel_data object
panel <- make_panel(production, index = c("firm", "year"))
plot_missing(panel)

# Selecting specific variables
plot_missing(production, select = c("labor", "capital"), index = c("firm", "year"))

# Custom colors
plot_missing(production, index = c("firm", "year"), colors = c("black", "white"))

# Access the returned list
out_plo_mis <- plot_missing(production, index = c("firm", "year"))
out_plo_mis$metadata
out_plo_mis$details
```

plot_patterns

Entities Presence Patterns Visualization

Description

This function creates a heatmap showing the presence/absence pattern of each entity over time.

Usage

```
plot_patterns(  
  data,  
  index = NULL,  
  delta = NULL,  
  limits = NULL,  
  colors = c("darkblue", "white")  
)
```

Arguments

| | |
|-------|--|
| data | A data.frame containing panel data in a long format. |
| index | A character vector of length 2 specifying the names of the entity and time variables. Not required if data has panel attributes. |

| | |
|--------|---|
| delta | An optional integer giving the expected interval between time periods. |
| limits | Either a single integer (show that many most frequent patterns) or a vector of two integers (show patterns with ranks between the two values, inclusive). If not specified, all patterns are shown. |
| colors | A character vector of two colors for present and missing observations. Default = c("darkblue", "white"). |

Details

The function creates a heatmap where rows are entities and columns are time periods. Present cells are colored with the first color, missing cells with the second. Rows are ordered by pattern frequency: the most frequent pattern is at the **top**. Within each pattern block, entities appear in their original order.

Effect of delta: If delta is supplied, the function checks for regular spacing and adds missing periods (with all zeros) to the plot. A message lists missing periods unless the interval was inherited from panel attributes. The heatmap will therefore show columns for the full regular time sequence, with missing periods appearing entirely white (or the color for missing).

The returned list contains:

metadata List containing the function name and the arguments used.

details List with the sorted presence matrix, pattern-entity mapping, pattern count, and the pattern matrix (unique patterns as rows).

Value

Invisibly returns a list with summary statistics and metadata.

Note

An entity-time combination is considered **present** if the corresponding row contains at least one non-NA value in any substantive variable (all columns except the entity and time identifiers).

See Also

See also [describe_patterns\(\)](#), [plot_periods\(\)](#), [plot_missing\(\)](#).

Examples

```
data(production)

# Basic usage
plot_patterns(production, index = c("firm", "year"))

# With panel_data object
panel <- make_panel(production, index = c("firm", "year"))
plot_patterns(panel)

# Specifying time interval
plot_patterns(production, index = c("firm", "year"), delta = 1)
```

```
# Show only the top 3 patterns
plot_patterns(production, index = c("firm", "year"), limits = 3)

# Show patterns ranked 4 to 6
plot_patterns(production, index = c("firm", "year"), limits = c(4, 6))

# Custom colors
plot_patterns(production, index = c("firm", "year"), colors = c("black", "white"))

# Accessing list components
out_plo_pat <- plot_patterns(production, index = c("firm", "year"))
out_plo_pat$metadata
out_plo_pat$details
```

plot_periods

Time Coverage Distribution Visualization

Description

This function calculates summary statistics and creates a histogram showing the distribution of time periods covered by each entity in panel data.

Usage

```
plot_periods(data, index = NULL, colors = c("darkblue", "white"))
```

Arguments

| | |
|--------|--|
| data | A data.frame containing panel data in a long format. |
| index | A character vector of length 2 specifying the names of the entity and time variables. Not required if data has panel attributes. |
| colors | A character vector of length 2 specifying the fill color and line color for the histogram. First color is for fill, second color is for the border line. Default = c("darkblue", "white"). |

Details

The function creates a histogram of the number of time periods covered by each entity. The x-axis shows coverage (periods per entity), the y-axis shows the count of entities.

The returned list contains:

metadata List containing the function name and the arguments used.

details List with the coverage vector per entity and the histogram data used for plotting.

Value

Invisibly returns a list with summary statistics and metadata.

Note

An entity-time combination is considered **present** if the corresponding row contains at least one non-NA value in any substantive variable (all columns except the entity and time identifiers).

See Also

See also [describe_periods\(\)](#), [plot_patterns\(\)](#), [plot_missing\(\)](#).

Examples

```
data(production)

# Basic usage
plot_periods(production, index = c("firm", "year"))

# With panel_data object
panel <- make_panel(production, index = c("firm", "year"))
plot_periods(panel)

# Custom colors
plot_periods(production, index = c("firm", "year"), colors = c("gray", "black"))

# Accessing list components
out_plo_per <- plot_periods(production, index = c("firm", "year"))
out_plo_per$metadata
out_plo_per$details
```

| | |
|------------|--|
| production | <i>Simulated Unbalanced Panel Data for Cobb-Douglas Production Function Analysis</i> |
|------------|--|

Description

A simulated dataset containing firm-level panel data with industry affiliation, entry, exit, random missing values, and ownership information. The data follows industry-specific production structures with occasional industry and ownership changes.

Usage

```
production
```

Format

A data frame with 180 rows (30 firms × 6 years) and 7 variables:

firm integer; firm identifier (1 to 30)

year integer; year identifier (1 to 6)

sales numeric; firm sales/output generated from a Cobb-Douglas production function with industry-specific parameters and technology shocks. Contains random missing values (~2%).

capital numeric; capital input, log-normally distributed with firm-specific effects and industry-specific time trends. Contains random missing values (~2%).

labor numeric; labor input, log-normally distributed with firm-specific effects and industry-specific time trends. Contains random missing values (~2%).

industry factor; industry affiliation with three levels: "Industry 1", "Industry 2", "Industry 3". Some firms change industry over time.

ownership factor; ownership type with three levels: "private", "public", "mixed". The variable is stable over time but changes with a probability of 5% per year.

Details

The dataset exhibits several realistic features of firm-level panel data:

- 50% of firms (15 firms) have complete data for all 6 years.
- 50% of firms (15 firms) have entry and exit patterns with different start and end years.
- Three industry categories with different production function parameters.
- About 20% of firms change industry affiliation at least once.
- Ownership changes occur with 5% probability per year.
- Industry-specific Cobb-Douglas parameters:
 - Industry 1: $\alpha = 0.25$, $\beta = 0.65$, $A = 2.0$ (labor-intensive)
 - Industry 2: $\alpha = 0.35$, $\beta = 0.55$, $A = 2.2$ (balanced, high productivity)
 - Industry 3: $\alpha = 0.30$, $\beta = 0.60$, $A = 1.8$ (standard)
- Additional random missing values (approx. 2%) in sales, capital, and labor.
- Firm-specific effects and industry-specific time trends in inputs.
- Technology shocks affecting output.

Source

Simulated data for econometric analysis and demonstration purposes.

Examples

```
data(production)
head(production)
table(production$ownership)
```

summarize_missing *Missing Values Summary for Panel Data*

Description

This function calculates summary statistics for missing values (NAs) in panel data, providing both overall and detailed period-specific missing value counts.

Usage

```
summarize_missing(
  data,
  select = NULL,
  index = NULL,
  detail = FALSE,
  digits = 3
)
```

Arguments

| | |
|--------|---|
| data | A data.frame containing panel data in a long format. |
| select | A character vector specifying which variables to analyze for missing values. If not specified, all variables (except entity and time) will be used. |
| index | A character vector of length 2 specifying the names of the entity and time variables. Not required if data has panel attributes. |
| detail | A logical flag indicating whether to return detailed period-specific NA counts. Default = FALSE. |
| digits | An integer indicating the number of decimal places to round the share column. Default = 3. |

Details

When `detail = FALSE`, returns columns:

variable Variable name.
na_count Total number of missing values in that variable.
na_share Proportion of missing values (rounded to digits).
entities Number of distinct entities that have at least one missing value in that variable.
periods Number of distinct time periods that have at least one missing value in that variable.

When `detail = TRUE`, additional columns for each time period contain the number of missing values in that variable for that period.

The object has class "panel_summary" and two additional attributes:

metadata List containing the function name and the arguments used.
details List with counts of variables with/without NAs, and their names.

Value

A data.frame with missing value summary statistics.

Note

The interpretation of missing counts may differ depending on whether the panel is balanced or unbalanced. In a balanced panel, each time period contains the same number of entities, so the raw NA counts per period are directly comparable across periods. In an unbalanced panel, the number of entities varies by period, so the raw NA counts should be interpreted relative to the number of observations available in each period. The function does not standardize the counts by period size; users should account for the panel structure when interpreting the results.

See Also

See also [plot_missing\(\)](#), [describe_incomplete\(\)](#), [describe_patterns\(\)](#), [describe_periods\(\)](#).

Examples

```
data(production)

# Basic usage
summarize_missing(production, index = c("firm", "year"))

# With panel_data object
panel <- make_panel(production, index = c("firm", "year"))
summarize_missing(panel)

# Selecting specific variables
summarize_missing(production, select = c("labor", "capital"), index = c("firm", "year"))

# Returning detailed results
summarize_missing(production, index = c("firm", "year"), detail = TRUE)

# Custom rounding
summarize_missing(production, index = c("firm", "year"), digits = 2)

# Accessing attributes
out_sum_mis <- summarize_missing(production, index = c("firm", "year"))
attr(out_sum_mis, "metadata")
attr(out_sum_mis, "details")
```

summarize_numeric

Summary Statistics for Numeric Variables

Description

This function calculates summary statistics for numeric variables, either overall or grouped by a single grouping variable.

Usage

```
summarize_numeric(  
  data,  
  select = NULL,  
  group = NULL,  
  detail = FALSE,  
  digits = 3  
)
```

Arguments

| | |
|---------------------|--|
| <code>data</code> | A <code>data.frame</code> containing variables for analysis. |
| <code>select</code> | A character vector specifying which numeric variables to analyze. If not specified, all numeric variables in the <code>data.frame</code> will be used. |
| <code>group</code> | A character string specifying the grouping variable name. If not specified, overall statistics will be returned. |
| <code>detail</code> | A logical flag indicating whether to return additional statistics (25th, 50th, and 75th percentiles). Default = <code>FALSE</code> . |
| <code>digits</code> | An integer specifying the number of decimal places for rounding statistics. Default = 3. |

Details

The returned `data.frame` contains columns depending on the arguments:

When no grouping variable is specified (overall):

| | |
|-----------------------|-----------------------------------|
| <code>variable</code> | The name of the numeric variable. |
| <code>count</code> | Number of non-NA observations. |
| <code>mean</code> | Arithmetic mean. |
| <code>std</code> | Standard deviation. |
| <code>min</code> | Minimum value. |
| <code>max</code> | Maximum value. |

When `detail = TRUE`, additional columns are included:

| | |
|------------------|-----------------------------------|
| <code>p25</code> | 25th percentile (first quartile). |
| <code>p50</code> | 50th percentile (median). |
| <code>p75</code> | 75th percentile (third quartile). |

When a grouping variable is specified, statistics are calculated for each group, and the `data.frame` includes a column named after the grouping variable, followed by the same statistics columns as above.

The object has class `"panel_summary"` and two additional attributes:

| | |
|-----------------------|--|
| <code>metadata</code> | List containing the function name and the arguments used. |
| <code>details</code> | List with counts of variables, groups, and total observations. |

Value

A data.frame with descriptive statistics summary.

See Also

See also [decompose_numeric\(\)](#), [plot_heterogeneity\(\)](#).

Examples

```
data(production)

# Basic usage
summarize_numeric(production)

# Selecting specific variables
summarize_numeric(production, select = "sales")
summarize_numeric(production, select = c("capital", "labor"))

# Grouped statistics
summarize_numeric(production, group = "year")

# Detailed statistics
summarize_numeric(production, detail = TRUE)

# Custom rounding
summarize_numeric(production, digits = 2)

# Accessing attributes
out_sum_num <- summarize_numeric(production)
attr(out_sum_num, "metadata")
attr(out_sum_num, "details")
```

summarize_transition *Transition Summary*

Description

Calculates transition counts and shares between states of a categorical (factor) variable across consecutive time periods within entities for panel data.

Usage

```
summarize_transition(data, select, index = NULL, format = "wide", digits = 3)
```

Arguments

| | |
|---------------------|--|
| <code>data</code> | A data.frame containing panel data in a long format. |
| <code>select</code> | A character string specifying the factor variable to analyze transitions for. |
| <code>index</code> | A character vector of length 2 specifying the names of the entity and time variables. Not required if data has panel attributes. |
| <code>format</code> | A character string specifying the output format: "wide" (default) or "long". |
| <code>digits</code> | An integer indicating the number of decimal places to round transition shares. Default = 3. |

Details

The structure depends on format:

When `format = "wide"`, a transition matrix as a data.frame:

`from_to` The originating state (row label).

`[state1]`, `[state2]`, ... Columns for each destination state, containing the share of transitions from the row state to the column state (rounded).

When `format = "long"`, a data.frame with columns:

`from` Originating state.

`to` Destination state.

`count` Number of observed transitions.

`share` Proportion of transitions from `from` that go to `to` (rounded).

The object has class "panel_summary" and two additional attributes:

`metadata` List containing the function name and the arguments used.

`details` List with the vector of all category levels.

Value

A data.frame containing transition summaries.

See Also

See also [decompose_factor\(\)](#).

Examples

```
data(production)

# Basic usage
summarize_transition(production, select = "industry", index = c("firm", "year"))

# With panel_data object
panel <- make_panel(production, index = c("firm", "year"))
summarize_transition(panel, select = "industry")
```

```
# Returning results in a long format
summarize_transition(production, select = "industry",
                    index = c("firm", "year"), format = "long")

# Custom rounding
summarize_transition(production, select = "industry", index = c("firm", "year"), digits = 2)

# Accessing attributes
out_sum_tra <- summarize_transition(production, select = "industry", index = c("firm", "year"))
attr(out_sum_tra, "metadata")
attr(out_sum_tra, "details")
```

Index

* datasets

- production, 22

- decompose_factor, 2
- decompose_factor(), 6, 28
- decompose_numeric, 4
- decompose_numeric(), 3, 17, 27
- describe_balance, 6
- describe_balance(), 9, 13, 14, 16
- describe_dimensions, 8
- describe_dimensions(), 8, 16
- describe_incomplete, 9
- describe_incomplete(), 25
- describe_patterns, 11
- describe_patterns(), 8, 10, 14, 20, 25
- describe_periods, 13
- describe_periods(), 8–10, 13, 16, 22, 25

- make_panel, 15

- plot_heterogeneity, 16
- plot_heterogeneity(), 6, 27
- plot_missing, 18
- plot_missing(), 20, 22, 25
- plot_patterns, 19
- plot_patterns(), 13, 19, 22
- plot_periods, 21
- plot_periods(), 8, 14, 19, 20
- production, 22

- summarize_missing, 19, 24
- summarize_missing(), 10
- summarize_numeric, 25
- summarize_numeric(), 6, 17
- summarize_transition, 27
- summarize_transition(), 3