

# Package: paneldesc (via r-universe)

May 30, 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

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Repository** <https://dtereshch.r-universe.dev>

**Date/Publication** 2026-05-30 17:18:57 UTC

**RemoteUrl** <https://github.com/dtereshch/paneldesc>

**RemoteRef** HEAD

**RemoteSha** 4735e74f40465d8436c813ac228efa4186d67616

## Contents

decompose_factor . . . . .	2
decompose_numeric . . . . .	4
describe_balance . . . . .	7
describe_dimensions . . . . .	9
describe_incomplete . . . . .	10

describe_patterns . . . . .	12
describe_periods . . . . .	14
make_balanced . . . . .	15
make_demeaned . . . . .	17
make_long . . . . .	19
make_panel . . . . .	20
make_wide . . . . .	22
plot_heterogeneity . . . . .	23
plot_missing . . . . .	24
plot_patterns . . . . .	26
plot_periods . . . . .	28
production . . . . .	29
summarize_missing . . . . .	30
summarize_numeric . . . . .	32
summarize_transition . . . . .	34

## Index 37

---

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	<i>Panel Data Numeric Variable Decomposition</i>
-------------------	--

---

## 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

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.
detail	A logical flag indicating whether to return additional statistics (5th, 25th, 50th, 75th, and 95th percentiles). Default = FALSE.
digits	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      *Entities Presence Patterns Description*

---

## 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

<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>delta</code>	An optional integer giving the expected interval between time periods.
<code>limits</code>	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.
<code>detail</code>	A logical flag indicating whether to return detailed patterns. Default = TRUE.
<code>format</code>	A character string specifying the output format: "wide" or "long". Default = "wide".
<code>digits</code>	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	<i>Time Periods Completeness Description</i>
------------------	--

---

## 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

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.
digits	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\_balanced

*Create a Balanced Panel Dataset*

---

### Description

This function creates a balanced panel dataset by either keeping only entities present in all time periods, keeping only periods where all entities are present, or expanding the data to include all entity-time combinations.

**Usage**

```
make_balanced(data, index = NULL, delta = NULL, balance = "rows")
```

**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". Specifies the balancing method (see Details). Default = "rows".

**Details**

This function balances a panel dataset according to the chosen method. The returned object has class "panel\_data" and includes metadata attributes similar to [make\\_panel\(\)](#).

**Balancing methods:**

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.

**Duplicates:** If duplicate entity-time combinations exist, the function stops with an error, as balancing requires a unique key.

**Missing values:** Rows with missing entity or time values are automatically removed before balancing.

**Handling of panel\_data objects:** If data is a panel\_data object, the function will use the entity, time, and delta values stored in its attributes unless overridden by explicit index or delta arguments.

**Value**

A balanced panel data.frame with additional attributes.

**See Also**

See also [make\\_panel\(\)](#), [make\\_wide\(\)](#), [make\\_long\(\)](#), [describe\\_dimensions\(\)](#), [describe\\_balance\(\)](#).

**Examples**

```
data(production)

# Create a panel object first
panel <- make_panel(production, index = c("firm", "year"))

# Expand to full grid (default method)
```

```

balanced_rows <- make_balanced(panel)

# Keep only entities present in all periods
balanced_entities <- make_balanced(panel, balance = "entities")

# Keep only periods where all entities are present
balanced_periods <- make_balanced(panel, balance = "periods")

# Using a regular data frame (index must be provided)
balanced_rows2 <- make_balanced(production, index = c("firm", "year"))

# Specifying time interval for yearly data
balanced_rows_delta <- make_balanced(production, index = c("firm", "year"), delta = 1)

```

---

make\_demeaned

*Within-Group Demeaning (Centering) for Panel Data*


---

## Description

This function performs within-group demeaning (centering) for all numeric variables in a data frame. For each group defined by the group argument, the group mean is subtracted from each observation. If no grouping is provided, the overall mean is subtracted (grand mean centering). Non-numeric variables are not demeaned and are returned unchanged.

## Usage

```
make_demeaned(data, group = NULL)
```

## Arguments

data	A data.frame containing the variables to be demeaned.
group	A character vector specifying the grouping variable(s). If not specified and data has panel attributes, the entity and time variables are used as grouping variables. Otherwise, overall demeaning is performed.

## Details

- If group is not specified and data is **not** a panel\_data object, simple overall demeaning is performed: for each numeric variable, the overall mean (ignoring NAs) is subtracted.
- If group is specified, the grouping variables are used to define the groups. Observations with NA in any grouping variable are removed before demeaning.
- **Missing values in numeric variables are not removed automatically**; the user should handle them prior to calling this function if desired.
- If data inherits from panel\_data and group is not specified, the function automatically uses the entity and time variables stored in the metadata attribute as grouping variables, and the returned object retains the panel\_data class and its attributes.

- Non-numeric variables are not demeaned and are returned unchanged.

#### Demeaning algorithms:

- One group:  $x - \text{mean}(x \mid \text{group})$  (exact, using `ave` with `na.rm = TRUE`).
- Two or more groups: iterative Gauss–Seidel algorithm (alternating projections). This matches the `fixest` fixed-effect residuals exactly, even for unbalanced panels. The algorithm runs up to 100 iterations with tolerance  $1e-12$ ; a warning is issued if convergence is not reached.

The returned object has a `metadata` attribute and a `details` attribute:

`metadata` List containing the function name ("make\_demeaned") and the grouping variables used (`group`). If the input was a `panel_data` object and `group` was not specified, the original panel metadata (`entity`, `time`, and `delta` if present) are also included.

`details` List with any additional information. If the input was a `panel_data` object, the original panel details are preserved.

#### Value

The input data frame with all numeric variables replaced by their demeaned versions. Rows with missing values in the grouping variables are removed. Missing values in numeric variables are left untouched, and group means are computed ignoring NAs.

#### See Also

See also [make\\_panel\(\)](#), [make\\_balanced\(\)](#), [make\\_wide\(\)](#), [make\\_long\(\)](#).

#### Examples

```
data(production)

# Simple overall demeaning
prod_demeaned <- make_demeaned(production)
head(prod_demeaned$labor)

# Demeaning by a single group (e.g., firm)
prod_demeaned_firm <- make_demeaned(production, group = "firm")

# Demeaning by two groups (e.g., firm and year) - matches fixest
prod_demeaned_both <- make_demeaned(production, group = c("firm", "year"))

# Using a panel_data object: automatically demeans by firm and year
panel <- make_panel(production, index = c("firm", "year"))
panel_demeaned <- make_demeaned(panel)
```

---

make_long	<i>Convert Panel Data from Wide to Long Format</i>
-----------	--

---

### Description

This function reshapes panel data from wide format to long format, stacking time-varying columns into rows based on the pattern of column names.

### Usage

```
make_long(data, index = NULL, spacer = "_", invert = FALSE)
```

### Arguments

data	A data.frame containing panel data in a wide format.
index	A character vector of length 2 specifying the name of the entity column (first element) and the name to give to the new time column in the long format (second element).
spacer	A character string used to separate variable names and time values in the wide column names. Default = "_".
invert	A logical flag indicating the order of components in column names. If FALSE (default), column names are "variable_spacer_time" (or "variable" + time when spacer = ""); if TRUE, they are "time_spacer_variable" (or time + "variable" when spacer = ""). Must match the structure of the input data.

### Details

The function performs the following steps:

- If data has panel attributes (e.g., from `make_wide()`) and `index` is not specified, the entity column, time column name, `spacer`, and `invert` are taken from the metadata.
- Columns that do not contain the `spacer` (or do not match the expected pattern when `spacer = ""`) are treated as time-constant and are replicated for each time period.
- Columns that match the pattern are split into variable names and time values; the set of unique time values defines the periods.
- The data are reshaped to long format using `stats::reshape()`.

The returned object has class "panel\_data" and two additional attributes:

`metadata` List containing the function name, the entity and time variables, the `spacer`, and the `invert` setting. If the input was a `panel_data` object, the original metadata elements (`delta`, etc.) are preserved.

`details` Preserved from the input if it was a `panel_data` object; otherwise an empty list.

### Value

A data frame in long format, with one row per entity-time combination.

**Note**

When `spacer = ""`, the function assumes that all time-varying columns have a numeric suffix (if `invert = FALSE`) or numeric prefix (if `invert = TRUE`) that represents the time period. Variable names may contain digits, but the last contiguous block of digits is treated as the time suffix; for prefixes, the first contiguous block of digits is treated as the time value. If a column does not contain any digit, it is considered time-constant.

The function assumes that all time-varying columns follow a consistent naming pattern and that every variable appears for exactly the same set of time periods (balanced in the wide sense). If some variable-time combinations are missing, a message is printed and those variables are omitted.

**See Also**

See also [make\\_panel\(\)](#), [make\\_wide\(\)](#), [make\\_balanced\(\)](#), [make\\_demeaned\(\)](#).

**Examples**

```
data(production)

# First convert to wide, then back to long
wide <- make_wide(production, index = c("firm", "year"))
long <- make_long(wide)
head(long)

# With custom spacer and invert
wide2 <- make_wide(production, index = c("firm", "year"), spacer = ".", invert = TRUE)
long2 <- make_long(wide2, spacer = ".", invert = TRUE)

# Using panel attributes (no need to specify index/spacer/invert)
panel <- make_panel(production, index = c("firm", "year"))
wide3 <- make_wide(panel)
long3 <- make_long(wide3)

# Using spacer = "" (no separator)
wide4 <- make_wide(production, index = c("firm", "year"), spacer = "")
long4 <- make_long(wide4, spacer = "")
```

---

make\_panel

*Panel Data Structure Setting*

---

**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.

**Usage**

```
make_panel(data, index, delta = 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.
...	Additional arguments (not used, except to catch deprecated balance).

## 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, and delta 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.

## Value

The input data.frame with additional attributes.

## See Also

See also [make\\_balanced\(\)](#), [make\\_balanced\(\)](#), [make\\_wide\(\)](#), [make\\_long\(\)](#), [make\\_demeaned\(\)](#), [describe\\_dimensions\(\)](#).

## 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)

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

---

 make\_wide
 

---



---

*Convert Panel Data from Long to Wide Format*


---

### Description

This function reshapes panel data from long format to wide format, creating separate columns for each time period.

### Usage

```
make_wide(data, index = NULL, spacer = "_", invert = FALSE)
```

### 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.
spacer	A character string to insert between variable names and time values in the wide format column names. Default = "_".
invert	A logical flag indicating whether to put time values before variable names in column names. If FALSE (default), column names are "variable_spacer_time"; if TRUE, they are "time_spacer_variable".

### Details

The function performs the following steps:

- If data has panel attributes and index is not specified, the entity and time variables are taken from the metadata.
- Rows with missing values in entity or time variables are removed.
- Duplicate entity-time combinations are detected and reported (unless they originate from panel attributes).
- The data are reshaped to wide format using `stats::reshape()`.

The returned object has class "panel\_data" and two additional attributes:

`metadata` List containing the function name, the entity and time variables, the spacer, and the invert setting. If the input was a panel\_data object, the original metadata elements (delta, etc.) are preserved.

`details` Preserved from the input if it was a panel\_data object; otherwise an empty list.

### Value

A data frame in wide format, with one row per entity.

**Note**

The function works for standard atomic types (logical, integer, double, complex, character, raw) and for factors. However, non-standard column types such as Date, POSIXct, or custom S3/S4 classes may lose their special attributes during reshaping. Duplicate entity-time combinations must be resolved beforehand; the function will issue a message but does not aggregate.

**See Also**

See also [make\\_panel\(\)](#), [make\\_long\(\)](#), [make\\_balanced\(\)](#), [make\\_demeaned\(\)](#).

**Examples**

```
data(production)

# Basic conversion
wide <- make_wide(production, index = c("firm", "year"))
head(wide)

# With panel_data object
panel <- make_panel(production, index = c("firm", "year"))
wide2 <- make_wide(panel)

# Custom spacer and inverted order
wide3 <- make_wide(production, index = c("firm", "year"),
                  spacer = ".", invert = TRUE)
names(wide3)
```

---

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 <b>largest</b> number of missing values, the second color the <b>smallest</b> 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  $\times$  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**

<code>data</code>	A data.frame containing panel data in a long format.
<code>select</code>	A character vector specifying which variables to analyze for missing values. If not specified, all variables (except entity and time) will be used.
<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 detailed period-specific NA counts. Default = FALSE.
<code>digits</code>	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, 29

decompose\_factor, 2  
decompose\_factor(), 6, 35  
decompose\_numeric, 4  
decompose\_numeric(), 3, 24, 34  
describe\_balance, 7  
describe\_balance(), 9, 13, 15, 16  
describe\_dimensions, 9  
describe\_dimensions(), 8, 16, 21  
describe\_incomplete, 10  
describe\_incomplete(), 32  
describe\_patterns, 12  
describe\_patterns(), 8, 11, 15, 27, 32  
describe\_periods, 14  
describe\_periods(), 8, 9, 11, 13, 29, 32

make\_balanced, 15  
make\_balanced(), 18, 20, 21, 23  
make\_demeaned, 17  
make\_demeaned(), 20, 21, 23  
make\_long, 19  
make\_long(), 16, 18, 21, 23  
make\_panel, 20  
make\_panel(), 16, 18, 20, 23  
make\_wide, 22  
make\_wide(), 16, 18, 20, 21

plot\_heterogeneity, 23  
plot\_heterogeneity(), 6, 34  
plot\_missing, 24  
plot\_missing(), 27, 29, 32  
plot\_patterns, 26  
plot\_patterns(), 13, 25, 29  
plot\_periods, 28  
plot\_periods(), 8, 15, 25, 27  
production, 29

summarize\_missing, 25, 30  
summarize\_missing(), 11  
summarize\_numeric, 32  
summarize\_numeric(), 6, 24  
summarize\_transition, 34  
summarize\_transition(), 3