Package: spreadview (via r-universe)

November 2, 2026

Title Quickly export survey results into MS Excel spreadsheet

Version 0.3.0

Description Export data (usually) from opinion survey into a nicely readable MS Excel spreadsheet. Based on data.table and openxlsx2. Supports categorical variables only for now.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports data.table, openxlsx2

Suggests testthat (>= 3.0.0)

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libicu-dev

Repository https://stem-cz.r-universe.dev

Date/Publication 2026-05-20 14:25:22 UTC

RemoteUrl https://github.com/alesvomacka/spreadview

RemoteRef HEAD

RemoteSha 9658eb8126166be9764099c4e8a653ca3a01965e

Contents

compose_spreadsheet 2
compose_table 3
compute_adj_residuals 4
count_freqs 6
get_categorical_vars 7

Index **8**

compose_spreadsheet *Compose a Spreadsheet with Multiple Frequency Tables*

Description

Creates an Excel workbook containing frequency tables for multiple variables, optionally grouped and weighted. Each variable's table is row-bound together into a single sheet. When grouping is specified, cells are colored based on adjusted standardized residuals.

Usage

```
compose_spreadsheet(
  data,
  vars,
  group = NULL,
  weight = NULL,
  file = NULL,
  na.rm = TRUE,
  percent = TRUE
)
```

Arguments

data	A data frame or data.table containing the variables to tabulate.
vars	Character vector of variable names to create tables for. Each variable will be passed to compose_table().
group	Character vector specifying grouping variables. Default is NULL for no grouping. Passed to compose_table().
weight	Character string naming the weight variable to use for weighted frequencies. Passed to compose_table().
file	Character string specifying the output file path. If NULL (default), returns the workbook object without saving.
na.rm	Should missing values be removed from the table? Defaults to TRUE.
percent	Logical. If TRUE (default), percentage values are formatted as percentages (e.g., 50\ as plain numbers multiplied by 100 (e.g., 50).

Details

The function iterates over each variable in vars, applies compose_table() with the specified parameters, and combines the results using data.table::rbindlist(). The resulting spreadsheet is added to a new workbook with openxlsx2.

The header spans two rows:

- Row 1: Column names for non-group columns (e.g., "label", "n", "total") and group variable names for grouped columns (e.g., "IDE_8")

- Row 2: Empty strings for non-group columns, category names for grouped columns (e.g., "muž", "žena")

When grouping variables are specified, cells are colored based on adjusted standardized residuals:

- Blue shades for positive residuals (more than expected)
- Red shades for negative residuals (less than expected)
- Color intensity increases with absolute residual value
- Residuals beyond ±2 indicate significance at ~0.05 level

Value

An `openxlsx2` workbook object containing a single worksheet with all frequency tables row-bound together. Cells are colored based on adjusted standardized residuals when grouping is used.

See Also

`compose_table`, `compute_adj_residuals`

Examples

```
## Not run:
compose_spreadsheet(data = penguins, var = c("species", "sex"), group = "island")

## End(Not run)
```

---

compose_table *Compose a frequency table with optional grouping*

Description

Creates a frequency table for a categorical variable, optionally cross-tabulated by one or more grouping variables. The function automatically adds variable labels and standardizes column names for easier downstream processing.

Usage

```
compose_table(
  data,
  var,
  group = NULL,
  weight = NULL,
  prop = "none",
  na.rm = TRUE
)
```

Arguments

data	A data frame or data.table containing the variables to tabulate.
var	Character string specifying the name of the variable to tabulate.
group	Character vector of variable name(s) to group by. If NULL (default), creates a simple frequency table. If provided, creates cross-tabulations with each grouping variable. Group columns are prefixed with [group_name] in the output.
weight	Character string specifying the name of the weight variable to use for weighted frequencies.
prop	Character string specifying proportion type. Options are "none" for counts only, "col" for column proportions, or "row" for row proportions. Default is "none".
na.rm	Should missing values be removed from the table? Defaults to TRUE.

Value

A data.table with the following structure:

item	Variable name of the tabulated variable
label	Variable label from the var attribute (empty string if not available)
var	Factor levels of the tabulated variable
n	Weighted counts
total	Overall proportions (if prop != "none")
\link{group} level	One column per level of each grouping variable (if group is specified)

Examples

```
compose_table(penguins, var = "species", group = "island", prop = "col")
```

---

compute_adj_residuals *Compute adjusted standardized residuals for contingency tables*

Description

Calculates adjusted standardized residuals for a contingency table formed by crossing a row variable with one or more grouping (column) variables. These residuals measure how much each observed cell frequency deviates from what would be expected under independence, adjusted for both row and column proportions.

Usage

```
compute_adj_residuals(data, var, group, weight)
```

Arguments

data	A data frame or tibble containing the survey data.
var	Character string specifying the row variable name.
group	Character vector specifying one or more column grouping variable names. When multiple groups are provided, residuals are computed separately for each group's contingency table.
weight	Character string specifying the weight variable name.

Details

Adjusted standardized residuals are calculated as:

$$\frac{O - E}{\sqrt{E \times (1 - p_{row}) \times (1 - p_{col})}}$$

where O is the observed frequency, E is the expected frequency under independence, p_{row} is the row proportion, and p_{col} is the column proportion.

These residuals can be interpreted like z-scores: values beyond ±2 suggest significant deviation from independence at approximately the 0.05 level. They are equivalent to the standardized residuals from `chisq.test()`.

When multiple grouping variables are provided, residuals are computed independently for each two-way contingency table (var × group_i), then combined into a single output table matching the structure of `compose_table()`.

Value

A data.table with the same structure as the output from `compose_table()` with the same grouping variables. The first column contains the row variable levels, and subsequent columns contain adjusted standardized residuals for each group level. When multiple groups are specified, columns are prefixed with `[group_name]` to match `compose_table()` output.

Examples

```
## Not run:
# Single group
compute_adj_residuals(cvvm, var = "t_VZD", group = "IDE_8", weight = "soc_weight")

# Multiple groups
compute_adj_residuals(
  cvvm,
  var = "t_VZD",
  group = c("t_VEK_6", "IDE_8"),
  weight = "soc_weight"
)

## End(Not run)
```

`count_freqs` *Calculate Frequency Tables with Optional Proportions*

Description

Computes weighted frequency tables for a variable, optionally grouped by another variable. Can return counts or proportions calculated across different margins (total, row, or column).

Usage

```
count_freqs(
  data,
  var,
  group = NULL,
  weight = NULL,
  prop = "none",
  na.rm = TRUE,
  include_empty = TRUE
)
```

Arguments

`data`	A data frame or object that can be coerced to a data.table.
`var`	Character string specifying the name of the variable to tabulate.
`group`	Character string specifying the name of the grouping variable, or `NULL` (default) for no grouping.
`weight`	Character string specifying the name of the weight variable to use for weighted frequencies. If no weights are specified, equal weights are assumed.
`prop`	Character string indicating the type of proportion to calculate. Options are: • `"none"` (default): Return weighted counts only • `"total"`: Proportions of the grand total • `"row"`: Proportions within each row (within each level of `var`) • `"col"`: Proportions within each column (within each level of `group`)
`na.rm`	Logical. If `TRUE`, removes rows with missing values in `var` before computing frequencies. Default is `FALSE`.
`include_empty`	Logical. If `TRUE` (default), factor levels of `var` (and `group`, if specified) with no observed values are included in the result with `n = 0`.

Value

A data.table containing:

- When `group` is `NULL`: A table with `var` and either `n` (counts) or `total` (proportions)
- When `group` is specified and `prop != "none"`: A wide-format table with `var` as rows and `group` levels as columns, containing proportions
- When `group` is specified and `prop == "none"`: A long-format table with `var`, `group`, and `n` columns

Examples

```
count_freqs(data = penguins, var = "species", prop = "none")
count_freqs(data = penguins, var = "species", group = "island", prop = "col")
```

`get_categorical_vars` *Get names of categorical variables from a data frame*

Description

Extracts the names of factor variables from a data frame. Optionally includes character variables and allows exclusion of specific variable names.

Usage

```
get_categorical_vars(data, exclude = NULL, include_char = FALSE)
```

Arguments

`data`	A data frame to extract categorical variable names from.
`exclude`	A character vector of variable names to exclude from the results. Default is `NULL` (no exclusions).
`include_char`	Logical. If `TRUE`, character variables are included in addition to factor variables. Default is `FALSE`.

Value

A character vector of variable names.

Examples

```
# Get only factor variables
get_categorical_vars(iris)

# Include character variables
get_categorical_vars(mtcars, include_char = TRUE)

# Exclude specific variables
get_categorical_vars(iris, exclude = c("Species"))
```

Index

compose_spreadsheet, 2
compose_table, 3, *3*
compute_adj_residuals, *3*, 4
count_freqs, 6

get_categorical_vars, 7