Package 'surveyexplorer'

December 21, 2023

Title Quickly Explore Complex Survey Data

Version 0.1.0

Description
Visualize and tabulate single-choice, multiple-choice, matrix-style questions from survey data. Includes ability to group cross-
tabulations, frequency distributions, and plots by categorical variables and
to integrate survey weights. Ideal for quickly uncovering descriptive patterns in survey data.

License MIT + file LICENSE

Depends R (>= 2.10)

Imports dplyr, ggplot2, ggupset, gt, purrr, rlang, scales, stringr,
tidyr, tidyselect

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

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Language en-US

NeedsCompilation no

Author Liam Haller [aut, cre, cph] (<https://orcid.org/0000-0001-8033-6599>)

Maintainer Liam Haller <liamhllr2@gmail.com>

Repository CRAN

Date/Publication 2023-12-21 16:40:02 UTC

R topics documented:

`berlinbears` *Bears bears dataframe*

Description

A "survey" of bears in Berlin Report ...

Usage

```
berlinbears
```

Format

`berlinbears`**:**

A data frame with 500 rows and 22 columns describing bears and their preferences:

species name of species

genus genus that the species belongs to

gender gender ofthe bear

age age of the bear

will_eat.SQ001, will_eat.SQ002, will_eat.SQ003, will_eat.SQ004, will_eat.SQ005 survey questions on foods the bear will eat

p_likespine, p_likeshoney, p_eatstrash, p_swims, p_hibernates,p_likes_zoo example of likert questions ...

`frequency_table` *Base table for single & multiple choice questions*

Description

Base table for single & multiple choice questions

Usage

```
frequency_table(data.table, group_by)
```

Arguments

data.table	Output from either mutli or single summary
group_by	Optional variable to group the analysis. If provided, the frequencies and counts will be calculated within each subgroup.

Value

Gt table

---

matrix_freq *Matrix Frequency Plot*

Description

Generate a grouped bar chart displaying the frequency distribution of responses for a categorical variable. The function supports optional subgrouping of data using the group_by variable, exclusion of specific subgroups with 'subgroups_to_exclude,' and data weighting with the 'weights' parameter. Users can also choose to exclude NA values from the questions prior to analysis using the 'na.rm' parameter.

Usage

```
matrix_freq(
  dataset,
  question,
  group_by = NULL,
  subgroups_to_exclude = NULL,
  weights = NULL,
  na.rm = FALSE
)
```

Arguments

dataset	The input dataframe (or tibble) of survey questions
question	The columns that contain each of the response options for a question, can be selected by using **tidyselect** semanatics or providing a vector of column names or numbers
group_by	Optional variable to group the analysis. If provided, the frequencies and counts will be calculated within each subgroup.
subgroups_to_exclude	Optional vector specifying subgroups to exclude from the analysis.
weights	Optional variable containing survey weights. If provided, frequencies and counts will be weighted accordingly.
na.rm	Logical indicating whether to remove NA values from question before analysis.

Value

A ggplot2 object representing a grouped bar chart displaying the frequency distribution of responses for the specified categorical variable. The chart supports grouping, weighting, and exclusion of subgroups.

See Also

Other matrix questions: `matrix_likert()`, `matrix_mean()`, `matrix_table()`

Examples

```
#Array question (1-5)
 matrix_freq(berlinbears, dplyr::starts_with('p_'))

 #remove NA category
 matrix_freq(berlinbears, dplyr::starts_with('p_'), na.rm = TRUE)

 #Use `group_by` to partition the question into several groups
 matrix_freq(berlinbears, dplyr::starts_with('p_'), group_by = species,
 subgroups_to_exclude = c('panda bear', NA ), na.rm = TRUE)

 #Categorical input
 matrix_freq(berlinbears, dplyr::starts_with('c_'), group_by = is_parent, na.rm = TRUE)
```

---

matrix_likert *Plot Likert-scale responses using ggplot2.*

Description

The function produces a visually appealing diverging stacked bar chart, allowing for easy interpretation of the distribution of responses to a specific Likert-scale question. The function supports customization of labels, colors, and weights, providing flexibility in data representation.

Usage

```
matrix_likert(
  dataset,
  question,
  labels = NULL,
  colors = NULL,
  weights = NULL,
  na.rm = TRUE
)
```

Arguments

`dataset`	The input dataframe (or tibble) of survey questions
`question`	The columns that contain each of the response options for a question, can be selected by using **tidyselect** semanatics or providing a vector of column names or numbers
`labels`	Optional vector specifying labels for each response category. If not provided, it extracts labels from the original dataset.
`colors`	Optional vector specifying colors for each response category. Default colors are provided for 3 and 5 categories. If not specified, the function expects a vector of color codes.
`weights`	Optional variable containing survey weights. If provided, frequencies and counts will be weighted accordingly.
`na.rm`	Logical indicating whether to remove NA values from `question` before analysis.

Value

A ggplot2 object representing a diverging stacked bar chart displaying the distribution of Likert-scale responses. The chart is customized based on the provided or extracted labels and colors.

See Also

Other matrix questions: `matrix_freq()`, `matrix_mean()`, `matrix_table()`

`matrix_mean` *Matrix Mean Plot*

Description

This function creates a likert-style plot showing means and standard errors for a specified numeric variable, `question`. Optionally, the plot can be grouped by another variable, `group_by`, and subgroups can be excluded. If survey weights are provided, the counts are adjusted accordingly. The plot is flipped for better readability in likert-style format.

Usage

```
matrix_mean(
  dataset,
  question,
  group_by = NULL,
  subgroups_to_exclude = NULL,
  weights = NULL,
  na.rm = FALSE
)
```

Arguments

dataset	The input dataframe (or tibble) of survey questions
question	The columns that contain each of the response options for a question, can be selected by using **tidyselect** semanatics or providing a vector of column names or numbers
group_by	Optional variable to group the analysis. If provided, the frequencies and counts will be calculated within each subgroup.
subgroups_to_exclude	Optional vector specifying subgroups to exclude from the analysis.
weights	Optional variable containing survey weights. If provided, frequencies and counts will be weighted accordingly.
na.rm	Logical indicating whether to remove NA values from question before analysis.

Value

A likert-style ggplot displaying means and standard errors. The plot is flipped for better readability, and if grouping is specified, different colors represent distinct subgroups.

See Also

Other matrix questions: matrix_freq(), matrix_likert(), matrix_table()

Examples

```
#basic plot
  matrix_mean(berlinbears, dplyr::starts_with('p_'))

 #with grouping and weights
   matrix_mean(berlinbears, dplyr::starts_with('p_'), group_by = species,
   subgroups_to_exclude = 'panda bear', weights = weights, na.rm = TRUE )
```

---

matrix_table *Create a table of frequencies and counts for matrix questions*

Description

This function creates a table showing percentages and counts for each response option in a multiple-choice question, specified by question. If grouping is provided with group_by, the table is extended to include subgroups. Subgroups can be excluded, and survey weights are supported for adjusted counts. The table is formatted for clarity and can be displayed in wide format. When weights are used, counts are presented as percentages only, and a note is added at the bottom of the table.

Usage

```
matrix_table(
  dataset,
  question,
  group_by = NULL,
  subgroups_to_exclude = NULL,
  weights = NULL,
  na.rm = FALSE
)
```

Arguments

dataset	The input dataframe (or tibble) of survey questions
question	The columns that contain each of the response options for a question, can be selected by using **tidyselect** semanatics or providing a vector of column names or numbers
group_by	Optional variable to group the analysis. If provided, the frequencies and counts will be calculated within each subgroup.
subgroups_to_exclude	Optional vector specifying subgroups to exclude from the analysis.
weights	Optional variable containing survey weights. If provided, frequencies and counts will be weighted accordingly.
na.rm	Logical indicating whether to remove NA values from question before analysis.

Value

A gt table summarizing percentages and counts for each response option in the specified multiple-choice question. If grouping is provided, the table includes subgroups and is formatted for clarity.

@examples #Array question (1-5) matrix_table(berlinbears, dplyr::starts_with('p_'))

#Use group_by to partition the question into several groups matrix_table(berlinbears, dplyr::starts_with('p_'), group_by = species, subgroups_to_exclude = 'panda bear')

#Remove NA category matrix_table(berlinbears, dplyr::starts_with('p_'), group_by = species, subgroups_to_exclude = 'panda bear', na.rm = TRUE

#Categorical input matrix_table(berlinbears, dplyr::starts_with('c_'), group_by = is_parent)

See Also

Other matrix questions: matrix_freq(), matrix_likert(), matrix_mean()

multi_freq *Generate an UpSet plot for multiple-choice questions*

Description

Visualize multiple-choice question responses with an upset plot, a visual tool for exploring the overlap and distribution of multiple-choice question responses. The function supports optional subgrouping of data using the group_by variable, exclusion of specific subgroups with 'subgroups_to_exclude,' and data weighting with the 'weights' parameter. Users can also choose to exclude NA values from the questions prior to analysis using the 'na.rm' parameter.

Usage

```
multi_freq(
  dataset,
  question,
  group_by = NULL,
  subgroups_to_exclude = NULL,
  weights = NULL,
  na.rm = FALSE
)
```

Arguments

dataset	The input dataframe (or tibble) of survey questions
question	The columns that contain each of the response options for a question, can be selected by using **tidyselect** semanatics or providing a vector of column names or numbers
group_by	Optional variable to group the analysis. If provided, the frequencies and counts will be calculated within each subgroup.
subgroups_to_exclude	Optional vector specifying subgroups to exclude from the analysis.
weights	Optional variable containing survey weights. If provided, frequencies and counts will be weighted accordingly.
na.rm	Logical indicating whether to remove NA values from question before analysis.

Value

An upset plot visualizing the distribution of responses to the multiple-choice question.

See Also

Other multiple-choice questions: multi_summary(), multi_table()

Examples

```
#Use dplyr to select questions
library(dplyr)

#Basic Upset plot

#Use `group_by` to partition the question into several groups
 multi_freq(berlinbears, question = dplyr::starts_with('will_eat'), group_by
 = gender)

#to ignore a subgroup, use `subgroups_to_exclude`
multi_freq(berlinbears, question = dplyr::starts_with('will_eat'), group_by =
gender, subgroups_to_exclude = NA)

#Specifiy survey weights with `weights`
 multi_freq(berlinbears, question = dplyr::starts_with('will_eat'), group_by
 = gender, weights = weights)
```

`multi_summary` *Compute summary statistics for a multiple choice questions*

Description

This function generates summary statistics, including frequencies, based on the provided question. It allows for optional grouping and weighting of data.

Usage

```
multi_summary(
  dataset,
  question,
  group_by = NULL,
  subgroups_to_exclude = NULL,
  weights = NULL,
  na.rm
)
```

Arguments

`dataset`	The input dataframe (or tibble) of survey questions
`question`	The columns that contain each of the response options for a question, can be selected by using **tidyselect** semanatics or providing a vector of column names or numbers

group_by	Optional variable to group the analysis. If provided, the frequencies and counts will be calculated within each subgroup.
subgroups_to_exclude	Optional vector specifying subgroups to exclude from the analysis.
weights	Optional variable containing survey weights. If provided, frequencies and counts will be weighted accordingly.
na.rm	Logical indicating whether to remove NA values from question before analysis.

Value

A data frame containing summary statistics, including frequencies, for the specified question.

See Also

Other multiple-choice questions: multi_freq(), multi_table()

---

multi_table *Create a table of frequencies and counts for multiple-choice questions*

Description

Generates a table presenting the distribution of responses for a specified multiple-choice question. If a grouping variable, group_by, is provided, the table extends to include row and column totals, along with additional count and frequency columns for each level of group_by (excluding specified subgroups, if any). When survey weights are specified with weights, the counts reflect the weighted values, and a note is appended at the bottom of the table.

Usage

```
multi_table(
  dataset,
  question,
  group_by = NULL,
  subgroups_to_exclude = NULL,
  weights = NULL,
  na.rm = FALSE
)
```

Arguments

dataset	The input dataframe (or tibble) of survey questions
question	The columns that contain each of the response options for a question, can be selected by using **tidyselect** semanatics or providing a vector of column names or numbers

group_by	Optional variable to group the analysis. If provided, the frequencies and counts will be calculated within each subgroup.
subgroups_to_exclude	Optional vector specifying subgroups to exclude from the analysis.
weights	Optional variable containing survey weights. If provided, frequencies and counts will be weighted accordingly.
na.rm	Logical indicating whether to remove NA values from question before analysis.

Value

A gt table displaying frequencies and counts for the specified multiple-choice question. If a grouping variable is provided, the table includes subgroups for a comprehensive analysis. If survey weights are specified, the table notes that frequencies and counts are weighted.

See Also

Other multiple-choice questions: multi_freq(), multi_summary()

Examples

```
#Basic Table
 multi_table(berlinbears, question = dplyr::starts_with('will_eat'))

#Use `group_by` to partition the question into several groups
 multi_table(berlinbears, question = dplyr::starts_with('will_eat'), group_by
 = gender)

#to ignore a subgroup, use `subgroups_to_exclude`
multi_table(berlinbears, question = dplyr::starts_with('will_eat'), group_by
= gender, subgroups_to_exclude = NA)

#Specifiy survey weights with `weights`
 multi_table(berlinbears, question = dplyr::starts_with('will_eat'), group_by
 = gender, weights = weights)
```

---

single_freq *Plot frequencies of responses for a single-choice question.*

Description

generates a bar chart of class ggplot illustrating how responses are distributed for a specific single-choice question. If you provide a grouping variable using group_by the chart includes facets for each subgroup. Additionally, if you specify survey weights with weights the chart reflects weighted response frequencies.

Usage

```
single_freq(
  dataset,
  question,
  group_by = NULL,
  subgroups_to_exclude = NULL,
  weights = NULL,
  na.rm = FALSE
)
```

Arguments

dataset	The input dataframe (or tibble) of survey questions
question	The categorical variable of interest for which frequencies and counts will be calculated, can be selected by using **tidyselect** semantics
group_by	Optional variable to group the analysis. If provided, the frequencies and counts will be calculated within each subgroup.
subgroups_to_exclude	Optional vector specifying subgroups to exclude from the analysis.
weights	Optional variable containing survey weights. If provided, frequencies and counts will be weighted accordingly.
na.rm	Logical indicating whether to remove NA values from question before analysis.

Value

A ggplot2 object with a bar chart displaying response frequencies. If "group_by" is provided, facets show subgroup details. If "weights" are specified, the chart displays weighted frequencies.

See Also

Other single-choice questions: single_summary(), single_table()

Examples

```
#Simple barchart
single_freq(berlinbears, question = income)

#Use `group_by` to facet the graph into several groups
single_freq(berlinbears, question = income, group_by = gender)

#to ignore a subgroup, use `subgroups_to_exclude`
single_freq(berlinbears, question = income, group_by = species,
subgroups_to_exclude = c('black bear', NA))

#Specify survey weights with `weights`
single_freq(berlinbears, question = h_winter, group_by = gender, weights = weights)
```

```
#to ignore NA values in the responses to `question`, set na.rm = TRUE
single_freq(berlinbears, question = h_winter, na.rm = TRUE)
```

`single_summary` *Generate a summary table for a single categorical variable, providing counts and frequencies.*

Description

This function analyzes a specified categorical variable, `question`, optionally grouping by another variable, `group_by`. Counts and frequencies are computed, taking into account provided survey weights. Subgroups can be excluded, and NAs can be removed if necessary.

Usage

```
single_summary(
  dataset,
  question,
  group_by = NULL,
  subgroups_to_exclude = NULL,
  weights = NULL,
  na.rm
)
```

Arguments

`dataset`	The input dataframe (or tibble) of survey questions
`question`	The categorical variable of interest for which frequencies and counts will be calculated, can be selected by using **tidyselect** semantics
`group_by`	Optional variable to group the analysis. If provided, the frequencies and counts will be calculated within each subgroup.
`subgroups_to_exclude`	Optional vector specifying subgroups to exclude from the analysis.
`weights`	Optional variable containing survey weights. If provided, frequencies and counts will be weighted accordingly.
`na.rm`	Logical indicating whether to remove NA values from `question` before analysis.

Value

A tabled data frame with counts and frequencies for the specified variable and optional grouping variable. The output is pre-processed, considering subgroup exclusions, NA removal, and survey weights if provided.

See Also

Other single-choice questions: `single_freq()`, `single_table()`

`single_table` *Create a table of frequencies and counts for single-choice questions*

Description

Generates a detailed table summarizing the frequencies and counts for each level of the specified variable, `question`. If a grouping variable, `group_by`, is provided, the table extends to include row and column totals, along with additional count and frequency columns for each level of `group_by` (excluding specified subgroups, if any). When survey weights are specified with `weights`, the counts reflect the weighted values, and a note is appended at the bottom of the table.

Usage

```
single_table(
  dataset,
  question,
  group_by = NULL,
  subgroups_to_exclude = NULL,
  weights = NULL,
  na.rm = FALSE
)
```

Arguments

`dataset`	The input dataframe (or tibble) of survey questions
`question`	The categorical variable of interest for which frequencies and counts will be calculated, can be selected by using **tidyselect** semantics
`group_by`	Optional variable to group the analysis. If provided, the frequencies and counts will be calculated within each subgroup.
`subgroups_to_exclude`	Optional vector specifying subgroups to exclude from the analysis.
`weights`	Optional variable containing survey weights. If provided, frequencies and counts will be weighted accordingly.
`na.rm`	Logical indicating whether to remove NA values from `question` before analysis.

Value

A gt table summarizing frequencies and counts based on the specified parameters. If the optional `group_by` parameter is provided, the output will be a grouped gt table, displaying frequencies and counts for each subgroup as well as row and column totals.

See Also

Other single-choice questions: single_freq(), single_summary()

Examples

```
#Simple table
single_table(berlinbears, question = income)

#Use `group_by` to partition the question into several groups
single_table(berlinbears, question = income, group_by = gender)

#to ignore a subgroup, use `subgroups_to_exclude`
single_table(berlinbears, question = income, group_by = species,
subgroups_to_exclude = c('black bear', NA))

#Specifiy survey weights with `weights`
 single_table(berlinbears, question = h_winter, group_by = gender,
 weights = weights)

#to ignore NA values in the responses to `question`, set na.rm = TRUE
single_table(berlinbears, question = h_winter, na.rm = TRUE)
```

Index