Package ‘volker’

February 22, 2024

Type Package
Title High-Level Functions for Tabulating, Charting and Reporting Survey Data
Version 1.0.2
Date 2024-02-03
Description Craft polished tables and plots in Markdown reports.
Simply choose whether to treat your data as counts or metrics,
and the package will automatically generate well-designed default tables and plots for you.
Boiled down to the basics, with labeling features and simple interactive reports.
All functions are ‘tidyverse’ compatible.
URL https://github.com/strohne/volker
BugReports https://github.com/strohne/volker/issues
License MIT + file LICENSE
Encoding UTF-8
RoxygenNote 7.2.3
LazyData true
Imports stats, rlang, lifecycle, glue, tibble, dplyr, tidyr,
tidyselect, ggplot2 (>= 2.2.1), scales, base64enc, stringr,
purrr, magrittr, forcats, skimr, knitr, kableExtra, rmarkdown,
janitor, psych
Depends R (>= 4.2)
Suggests tidyverse, remote, usethis, testthat (>= 3.0.0)
VignetteBuilder knitr
Config/testthat/edition 3
NeedsCompilation no
Author Jakob Jünger [aut, cre, cph] (<https://orcid.org/0000-0003-1860-6695>),
Henrieke Kotthoff [ctb],
Chantal Gärtner [ctb] (<https://orcid.org/0000-0002-3653-6013>)
Maintainer Jakob Jünger <jakob.juenger@uni-muenster.de>
Repository CRAN
Date/Publication 2024-02-22 21:00:02 UTC
Description
A small random subset of data from a survey about ChatGPT adoption. The survey was conducted in April 2023 within the population of Germany Internet users.

Usage
chatgpt

Format
chatgpt:
A data frame with 101 rows and 19 columns:
case A running case number
adopter Adoption groups inspired by Roger’s innovator typology.
use_ Columns starting with use contain data about ChatGPT usage in different contexts.
cg_activities Text answers to the question, what the respondents do with ChatGPT.
cg_adoption_ A scale consisting of items about advantages, fears, and social aspects. The scales match theoretical constructs inspired by Roger’s diffusion model and Davis’ Technology Acceptance Model
sd_ Columns starting with sd contain sociodemographics of the respondents.

Details
Call codebook(volk::chatgpt) to see the items and answer options.
Source

Communication Department of the University of Münster (gehrau@uni-muenster.de).

codebook

Get variable labels from their comment attributes

Description

Get variable labels from their comment attributes

Usage

codebook(data, cols)

Arguments

data A tibble
cols A tidy variable selections to filter specific columns

Value

A tibble with the columns: - item_name: The column name. - item_group: First part of the column name, up to an underscore. - item_class: The last class value of an item (e.g. numeric, factor). - item_label: The comment attribute of the column. - value_name: In case a column has numeric attributes, the attribute names - value_label: In case a column has numeric attributes or T/F-attributes, the attribute values. In case a column has a levels attribute, the levels.

Examples

volker::codebook(volker::chatgpt)

data_clean

Prepare dataframe for tabs, plots, and index operations

Description

The tibble remembers whether it was already cleaned and the cleaning plan is only performed once in the first call.

Usage

data_clean(data, plan = "sosci", ...)

Arguments

- **data** Data frame
- **plan** The cleaning plan. By now, only "sosci" is supported. See `data_clean_sosci`.
- ... Other parameters passed to the appropriate cleaning function

Value

Cleaned data frame with vlkr_df class

Examples

```r
ds <- volker::chatgpt
ds <- data_clean(ds)
```

---

**html_report**  
*Volker style HTML document format*

Description

Based on the standard theme, tweaks the pill navigation to switch between tables and plots. To use the format, in the header of your Markdown document, set `output: volker::html_report`.

Usage

```r
html_report(...) 
```

Arguments

... Additional arguments passed to `html_document`

Value

R Markdown output format

Examples

```r
## Not run:
# Add 'volker::html_report' to the output options of your Markdown document:
#
# ```
# ---
# title: "How to create reports?"
# output: volker::html_report
# ---
# ```
#
## End(Not run)
```
### idx_add

Calculate the mean value of multiple items

**Description**

Calculate the mean value of multiple items

**Usage**

```r
idx_add(data, cols, newcol = NULL, negative = FALSE, clean = TRUE)
```

**Arguments**

- **data**: A dataframe
- **cols**: A tidy selection of item columns
- **newcol**: Name of the index as a character value. Set to NULL (default) to automatically build a name from the common column prefix, prefixed with "idx_"
- **negative**: If FALSE (default), negative values are recoded as missing values.
- **clean**: Prepare data by `data_clean`.

**Value**

The input tibble with an additional column that contains the index values. The column contains the result of the alpha calculation in the attribute named "psych.alpha".

**Examples**

```r
ds <- volker::chatgpt
volker::idx_add(ds, starts_with("cg_adoption"))
```

---

### labs_apply

Set variable labels by setting their comment attributes

**Description**

Set variable labels by setting their comment attributes

**Usage**

```r
labs_apply(data, codes, cols = NULL, values = TRUE)
```
Arguments

- **data**: A tibble
- **codes**: A tibble in codebook format. To set column labels, use item_name and item_label columns.
- **cols**: A tidy column selection. Set to NULL (default) to apply to all columns found in the codebook. Restricting the columns is helpful when you want to set value labels. In this case, provide a tibble with value_name and value_label columns and specify the columns that should be modified.
- **values**: If TRUE (default), sets value labels. - For factors: Factor levels and order are retrieved from the value_label column. - For item values: they are retrieved from both the columns value_name and value_label in your codebook.

Value

A tibble with new labels

Examples

```r
library(tibble)
library(volker)

newlabels <- tribble(
  ~item_name, ~item_label,
  "cg_adoption_advantage_01", "Allgemeine Vorteile",
  "cg_adoption_advantage_02", "Finanzielle Vorteile",
  "cg_adoption_advantage_03", "Vorteile bei der Arbeit",
  "cg_adoption_advantage_04", "Macht mehr Spaß"
)

tab_metrics(starts_with("cg_adoption_advantage_"))
```

### labs_clear

Remove all comments from the selected columns

#### Usage

`labs_clear(data, cols, labels = NULL)`
### labs_restore

**Arguments**

- `data` A tibble
- `cols` Tidyselect columns
- `labels` The attributes to remove. NULL to remove all attributes except levels and class

**Value**

A tibble with comments removed

**Examples**

```r
library(volker)
volker::chatgpt |> 
   labs_clear()
```

**Description**

You can store labels before mutate operations by calling `labs_store`.

**Usage**

`labs_restore(data, cols = NULL, values = TRUE)`

**Arguments**

- `data` A data frame
- `cols` A tidyselect column selection
- `values` If TRUE (default), restores value labels in addition to item labels. Item labels correspond to columns, value labels to values in the columns.

**Value**

A data frame

**Examples**

```r
columns <- c("sd_age")
library(dplyr)
library(volker)

volker::chatgpt |> 
  labs_store() |> 
  mutate(sd_age = 2024 - sd_age) |> 
  labs_restore() |> 
  tab_metrics(sd_age)
```
### labs_store

*Get the current codebook and store it in the codebook attribute.*

**Description**

You can restore the labels after mutate operations by calling `labs_restore`.

**Usage**

```r
labs_store(data)
```

**Arguments**

- `data` A data frame

**Value**

A data frame

**Examples**

```r
library(dplyr)
library(volker)

volker::chatgpt |>
  labs_store() |>
  mutate(sd_age = 2024 - sd_age) |>
  labs_restore() |>
  tab_metrics(sd_age)
```

---

### plot_counts

*Output a frequency plot*

**Description**

The type of frequency plot depends on the number of selected columns:

- One column: see `plot_counts_one`
- Multiple columns: see `plot_counts_items`
- One column and one grouping column: see `plot_counts_one_grouped`
- Multiple columns and one grouping column: see `plot_counts_items_grouped`

**Usage**

```r
plot_counts(data, cols, col_group = NULL, clean = TRUE, ...)
```
### plot_metrics

**Arguments**

- **data**  
  A data frame
- **cols**  
  A tidy column selection, e.g. a single column (without quotes) or multiple columns selected by methods such as `starts_with()`
- **col_group**  
  Optional, a grouping column. The column name without quotes.
- **clean**  
  Prepare data by `data_clean`.
- **...**  
  Other parameters passed to the appropriate plot function

**Value**

A `ggplot2` plot object

**Examples**

```r
library(volker)
data <- volker::chatgpt

plot_counts(data, sd_gender)
```

---

### Description

The table type depends on the number of selected columns:

- One column: see `plot_metrics_one`
- Multiple columns: see `plot_metrics_items`
- One column and one grouping column: see `plot_metrics_one_grouped`
- Multiple columns and one grouping column: see `plot_metrics_items_grouped`

**Usage**

```r
plot_metrics(data, cols, col_group = NULL, clean = TRUE, ...)
```

**Arguments**

- **data**  
  A data frame
- **cols**  
  A tidy column selection, e.g. a single column (without quotes) or multiple columns selected by methods such as `starts_with()`.
- **col_group**  
  Optional, a grouping column (without quotes).
- **clean**  
  Prepare data by `data_clean`.
- **...**  
  Other parameters passed to the appropriate plot function
Value
A ggplot object

Examples
library(volker)
data <- volker::chatgpt

plot_metrics(data, sd_age)

report_counts
Create table and plot for categorical variables

Description
Depending on your column selection, different types of plots and tables are generated. See plot_counts and tab_counts.

Usage
report_counts(
data, 
cols, 
col_group = NULL, 
index = TRUE, 
numbers = NULL, 
title = TRUE, 
close = TRUE, 
clean = TRUE, 
...
)

Arguments
data A data frame
cols A tidy column selection, e.g., a single column (without quotes) or multiple columns selected by methods such as starts_with().
col_group Optional, a grouping column (without quotes).
index When the cols contain items on a metric scale (as determined by get_direction), an index will be calculated using the 'psych' package. Set to FALSE to suppress index generation.
numbers The numbers to print on the bars: "n" (frequency), "p" (percentage) or both. Set to NULL to remove numbers.
title A character providing the heading or TRUE (default) to output a heading. Classes for tabset pills will be added.
**report_metrics**

Create table and plot for metric variables

---

**Description**

Depending on your column selection, different types of plots and tables are generated. See `plot_metrics` and `tab_metrics`.

**Usage**

```r
report_metrics(
  data,
  cols,
  col_group = NULL,
  ...,
  index = TRUE,
  title = TRUE,
  close = TRUE,
  clean = TRUE
)
```

**close**

Whether to close the last tab (default value TRUE) or to keep it open. Keep it open to add further custom tabs by adding headers on the fifth level in Markdown (e.g. ####### Method)

**clean**

Prepare data by `data_clean`.

... Parameters passed to the plot and tab functions.

**Details**

For item batteries, an index is calculated and reported. When used in combination with the Markdown-template "html_report", the different parts of the report are grouped under a tabsheet selector.

**Value**

A volker report object

**Examples**

```r
library(volker)
data <- volker::chatgpt

report_counts(data, sd_gender)
```
Arguments

- **data**: A data frame
- **cols**: A tidy column selection, e.g. a single column (without quotes) or multiple columns selected by methods such as `starts_with()`.
- **col_group**: Optional, a grouping column (without quotes).
- **...**: Parameters passed to the `plot` and `tab` functions.
- **index**: When the cols contain items on a metric scale (as determined by `get_direction`), an index will be calculated using the 'psych' package. Set to FALSE to suppress index generation.
- **title**: A character providing the heading or TRUE (default) to output a heading. Classes for tabset pills will be added.
- **close**: Whether to close the last tab (default value TRUE) or to keep it open. Keep it open to add further custom tabs by adding headers on the fifth level in Markdown (e.g. ###### Method)
- **clean**: Prepare data by `data_clean`.

Details

For item batteries, an index is calculated and reported. When used in combination with the Markdown-template 'html_report', the different parts of the report are grouped under a tabsheet selector.

Value

A volker report object

Examples

```r
library(volker)
data <- volker::chatgpt
report_metrics(data, sd_age)
```

Description

The type of frequency table depends on the number of selected columns:

- One column: see `tab_counts_one`
- Multiple columns: see `tab_counts_items`
- One column and one grouping column: see `tab_counts_one_grouped`
- Multiple columns and one grouping column: see `tab_counts_items_grouped`
**Usage**

```r
tab_counts(data, cols, col_group = NULL, clean = TRUE, ...)
```

**Arguments**

- `data` A data frame
- `cols` A tidy column selection, e.g. a single column (without quotes) or multiple columns selected by methods such as `starts_with()`
- `col_group` Optional, a grouping column. The column name without quotes.
- `clean` Prepare data by `data_clean`
- `...` Other parameters passed to the appropriate table function

**Value**

A volker tibble

**Examples**

```r
library(volker)
data <- volker::chatgpt
tab_counts(data, sd_gender)
```

---

**Description**

The table type depends on the number of selected columns:

- One column: see `tab_metrics_one`
- Multiple columns: see `tab_metrics_items`
- One column and one grouping column: see `tab_metrics_one_grouped`
- Multiple columns and one grouping column: see `tab_metrics_items_grouped`

**Usage**

```r
tab_metrics(data, cols, col_group = NULL, clean = TRUE, ...)
```

**Arguments**

- `data` A data frame
- `cols` A tidy column selection, e.g. a single column (without quotes) or multiple columns selected by methods such as `starts_with()`.
- `col_group` Optional, a grouping column (without quotes).
- `clean` Prepare data by `data_clean`
- `...` Other parameters passed to the appropriate table function
Value

A volker tibble

Examples

```r
library(volker)
data <- volker::chatgpt

tab_metrics(data, sd_age)
```
Index

* datasets
  chatgpt, 2
  codebook, 3, 6
  data_clean, 3, 5, 9, 11–13
  data_clean_sosci, 4
  get_direction, 10, 12
  html_report, 4
  idx_add, 5
  labs_apply, 5
  labs_clear, 6
  labs_restore, 7, 8
  labs_store, 7, 8
  plot_counts, 8, 10
  plot_counts_items, 8
  plot_counts_items_grouped, 8
  plot_counts_one, 8
  plot_counts_one_grouped, 8
  plot_metrics, 9, 11
  plot_metrics_items, 9
  plot_metrics_items_grouped, 9
  plot_metrics_one, 9
  plot_metrics_one_grouped, 9
  report_counts, 10
  report_metrics, 11
  tab_counts, 10, 12
  tab_counts_items, 12
  tab_counts_items_grouped, 12
  tab_counts_one, 12
  tab_counts_one_grouped, 12
  tab_metrics, 11, 13
  tab_metrics_items, 13