

Package ‘volker’

May 9, 2024

Type Package

Title High-Level Functions for Tabulating, Charting and Reporting
Survey Data

Version 2.0.0

Date 2024-05-07

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 7.3.1

LazyData true

Imports stats, rlang, lifecycle, tibble, dplyr, tidyr, tidyselect,
ggplot2 (>= 2.2.1), scales, base64enc, purrr, magrittr, skimr,
broom, knitr, kableExtra, rmarkdown, psych, effectsize, car

Depends R (>= 4.2)

Suggests tidyverse, remotes, 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 [aut, ctb],
Chantal Gärtner [ctb] (<<https://orcid.org/0000-0002-3653-6013>>)

Maintainer Jakob Jünger <jakob.juenger@uni-muenster.de>

Repository CRAN

Date/Publication 2024-05-09 20:30:03 UTC

R topics documented:

| | |
|--------------------------|-----------|
| chatgpt | 2 |
| codebook | 3 |
| effect_counts | 4 |
| effect_metrics | 5 |
| html_report | 6 |
| idx_add | 6 |
| labs_apply | 7 |
| labs_clear | 8 |
| labs_restore | 9 |
| labs_store | 10 |
| plot_counts | 10 |
| plot_metrics | 11 |
| report_counts | 12 |
| report_metrics | 14 |
| tab_counts | 15 |
| tab_metrics | 16 |
| theme_vlkr | 17 |
| Index | 19 |

chatgpt

ChatGPT Adoption Dataset CG-GE-APR22

Description

A small random subset of data from a survey about ChatGPT adoption. The survey was conducted in April 2023 within the population of German 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(volker::chatgpt)` to see the items and answer options.

Source

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

| | |
|----------|--|
| codebook | <i>Get variable labels from their comment attributes</i> |
|----------|--|

Description

[Experimental]

Usage

```
codebook(data, cols)
```

Arguments

| | |
|-------------------|--|
| <code>data</code> | A tibble. |
| <code>cols</code> | 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)
```

effect_counts *Output effect sizes for count data*

Description

The type of effect size depends on the number of selected columns:

- One column: see [effect_counts_one](#) (not yet implemented)
- Multiple columns: see [effect_counts_items](#) (not yet implemented)
- One column and one grouping column: see [effect_counts_one_grouped](#)
- Multiple columns and one grouping column: see [effect_counts_items_grouped](#) (not yet implemented)

By default, if you provide two column selections, the second column is treated as categorical. Setting the metric-parameter to TRUE will call the appropriate functions for correlation analysis:

- One column and one metric column: see [effect_counts_one_cor](#) (not yet implemented)
- Multiple columns and one metric column: see [effect_counts_items_cor](#) (not yet implemented)

[Experimental]

Usage

```
effect_counts(data, cols, cross = NULL, metric = FALSE, 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 <code>starts_with()</code> . |
| cross | Optional, a grouping column. The column name without quotes. |
| metric | When crossing variables, the cross column parameter can contain categorical or metric values. By default, the cross column selection is treated as categorical data. Set metric to TRUE, to treat it as metric and calculate correlations. |
| clean | Prepare data by data_clean . |
| ... | Other parameters passed to the appropriate effect function. |

Value

A volker tibble.

Examples

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

effect_counts(data, sd_gender, adopter)
```

| | |
|----------------|--|
| effect_metrics | <i>Output effect sizes and regression model parameters</i> |
|----------------|--|

Description

The regression type depends on the number of selected columns:

- One column: see [effect_metrics_one](#) (not yet implemented)
- Multiple columns: see [effect_metrics_items](#)
- One column and one grouping column: see [effect_metrics_one_grouped](#)
- Multiple columns and one grouping column: see [effect_metrics_items_grouped](#) (not yet implemented)

By default, if you provide two column selections, the second column is treated as categorical. Setting the metric-parameter to TRUE will call the appropriate functions for correlation analysis:

- Two metric columns: see [effect_metrics_one_cor](#)
- Multiple columns: see [effect_metrics_items_cor](#)

[Experimental]

Usage

```
effect_metrics(data, cols, cross = NULL, metric = FALSE, 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 <code>starts_with()</code> . |
| cross | Optional, a grouping column (without quotes). |
| metric | When crossing variables, the cross column parameter can contain categorical or metric values. By default, the cross column selection is treated as categorical data. Set metric to TRUE, to treat it as metric and calculate correlations. |
| clean | Prepare data by data_clean . |
| ... | Other parameters passed to the appropriate effect function. |

Value

A volker tibble.

Examples

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

effect_metrics(data, sd_age, sd_gender)
```

`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

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

Arguments

`...` Additional arguments passed to `html_document`.

Value

R Markdown output format.

Examples

```
## 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

[Experimental]

Usage

```
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

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

| | |
|------------|--|
| labs_apply | <i>Set variable labels by setting their comment attributes</i> |
|------------|--|

Description

[Experimental]

Usage

```
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

```
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"
)

volker::chatgpt %>%
  labs_apply(newlabels) %>%
  tab_metrics(starts_with("cg_adoption_advantage_"))
```

labs_clear

Remove all comments from the selected columns

Description

[Experimental]

Usage

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

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

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

| | |
|--------------|--|
| labs_restore | <i>Restore labels from the codebook store in the codebook attribute.</i> |
|--------------|--|

Description

[Experimental]

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. |

Details

You can store labels before mutate operations by calling [labs_store](#).

Value

A data frame.

Examples

```
library(dplyr)
library(volker)

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

| | |
|------------|---|
| labs_store | <i>Get the current codebook and store it in the codebook attribute.</i> |
|------------|---|

Description

You can restore the labels after mutate operations by calling [labs_restore](#).

Usage

```
labs_store(data)
```

Arguments

data A data frame.

Details

[Experimental]

Value

A data frame.

Examples

```
library(dplyr)
library(volker)

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

| | |
|-------------|--------------------------------|
| plot_counts | <i>Output a frequency plot</i> |
|-------------|--------------------------------|

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](#) (not yet implemented)

By default, if you provide two column selections, the second column is treated as categorical. Setting the metric-parameter to TRUE will call the appropriate functions for correlation analysis:

- One column and one metric column: see [plot_counts_one_cor](#) (not yet implemented)
- Multiple columns and one metric column: see [plot_counts_items_cor](#) (not yet implemented)

[Experimental]

Usage

```
plot_counts(data, cols, cross = NULL, metric = FALSE, 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 <code>starts_with()</code> . |
| cross | Optional, a grouping column. The column name without quotes. |
| metric | When crossing variables, the cross column parameter can contain categorical or metric values. By default, the cross column selection is treated as categorical data. Set metric to TRUE, to treat it as metric and calculate correlations. |
| clean | Prepare data by data_clean . |
| ... | Other parameters passed to the appropriate plot function. |

Value

A ggplot2 plot object.

Examples

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

plot_counts(data, sd_gender)
```

plot_metrics

Output a plot with distribution parameters such as the mean values

Description

The plot 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](#)

By default, if you provide two column selections, the second column is treated as categorical. Setting the metric-parameter to TRUE will call the appropriate functions for correlation analysis:

- Two metric columns: see [plot_metrics_one_cor](#)
- Multiple columns: see [plot_metrics_items_cor](#) (not yet implemented)

[Experimental]

Usage

```
plot_metrics(data, cols, cross = NULL, metric = FALSE, 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 <code>starts_with()</code> . |
| cross | Optional, a grouping column (without quotes). |
| metric | When crossing variables, the cross column parameter can contain categorical or metric values. By default, the cross column selection is treated as categorical data. Set metric to TRUE, to treat it as metric and calculate correlations. |
| 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,
  cross = NULL,
  metric = FALSE,
  index = FALSE,
  effect = FALSE,
  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 <code>starts_with()</code> . |
| cross | Optional, a grouping column (without quotes). |
| metric | When crossing variables, the cross column parameter can contain categorical or metric values. By default, the cross column selection is treated as categorical data. Set <code>metric</code> to <code>TRUE</code> , to treat it as metric and calculate correlations. |
| index | When the cols contain items on a metric scale (as determined by <code>get_direction</code>), an index will be calculated using the 'psych' package. Set to <code>FALSE</code> to suppress index generation. |
| effect | Whether to report statistical tests and effect sizes. See <code>effect_counts</code> for further parameters. |
| numbers | The numbers to print on the bars: "n" (frequency), "p" (percentage) or both. Set to <code>NULL</code> to remove numbers. |
| title | A character providing the heading or <code>TRUE</code> (default) to output a heading. Classes for tabset pills will be added. |
| close | Whether to close the last tab (default value <code>TRUE</code>) 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 <code>data_clean</code> . |
| ... | Parameters passed to the <code>plot_counts</code> and <code>tab_counts</code> and <code>effect_counts</code> 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.

[Experimental]

Value

A volker report object.

Examples

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

report_counts(data, sd_gender)
```

| | |
|----------------|---|
| report_metrics | <i>Create table and plot for metric variables</i> |
|----------------|---|

Description

Depending on your column selection, different types of plots and tables are generated. See [plot_metrics](#) and [tab_metrics](#).

Usage

```
report_metrics(
  data,
  cols,
  cross = NULL,
  metric = FALSE,
  ...,
  index = FALSE,
  effect = FALSE,
  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 <code>starts_with()</code> . |
| cross | Optional, a grouping or correlation column (without quotes). |
| metric | When crossing variables, the cross column parameter can contain categorical or metric values. By default, the cross column selection is treated as categorical data. Set <code>metric</code> to <code>TRUE</code> , to treat it as metric and calculate correlations. |
| ... | Parameters passed to the plot_metrics and tab_metrics and effect_metrics 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. |
| effect | Whether to report statistical tests and effect sizes. See effect_counts for further parameters. |
| 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.

[Experimental]

Value

A volker report object.

Examples

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

report_metrics(data, sd_age)
```

| | |
|------------|---------------------------------|
| tab_counts | <i>Output a frequency table</i> |
|------------|---------------------------------|

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](#)

By default, if you provide two column selections, the second column is treated as categorical. Setting the metric-parameter to TRUE will call the appropriate functions for correlation analysis:

- One column and one metric column: see [tab_counts_one_cor](#) (not yet implemented)
- Multiple columns and one metric column: see [tab_counts_items_cor](#) (not yet implemented)

[Experimental]

Usage

```
tab_counts(data, cols, cross = NULL, metric = FALSE, 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 <code>starts_with()</code> . |
| cross | Optional, a grouping column. The column name without quotes. |
| metric | When crossing variables, the cross column parameter can contain categorical or metric values. By default, the cross column selection is treated as categorical data. Set metric to TRUE, to treat it as metric and calculate correlations. |
| clean | Prepare data by data_clean . |
| ... | Other parameters passed to the appropriate table function. |

Value

A volker tibble.

Examples

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

tab_counts(data, sd_gender)
```

tab_metrics

Output a table with distribution parameters

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](#)

By default, if you provide two column selections, the second column is treated as categorical. Setting the metric-parameter to TRUE will call the appropriate functions for correlation analysis:

- Two metric columns: see [tab_metrics_one_cor](#)
- Multiple columns: see [tab_metrics_items_cor](#) (experimental)

[Experimental]

Usage

```
tab_metrics(data, cols, cross = NULL, metric = FALSE, 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 <code>starts_with()</code> . |
| cross | Optional, a grouping column (without quotes). |
| metric | When crossing variables, the cross column parameter can contain categorical or metric values. By default, the cross column selection is treated as categorical data. Set metric to TRUE, to treat it as metric and calculate correlations. |
| clean | Prepare data by data_clean . |
| ... | Other parameters passed to the appropriate table function. |

Value

A volker tibble.

Examples

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

tab_metrics(data, sd_age)
```

```
theme_vlkr
```

Define a default theme for volker plots

Description

Set ggplot colors, sizes and layout parameters.

Usage

```
theme_vlkr(
  base_size = 11,
  base_color = "black",
  base_fill = VLKR_FILLDISCRETE,
  base_gradient = VLKR_FILLGRADIENT
)
```

Arguments

| | |
|----------------------------|--|
| <code>base_size</code> | Base font size. |
| <code>base_color</code> | Base font color. |
| <code>base_fill</code> | A list of fill color sets or at least one fill color set. Example: <code>list(c("red"), c("red", "blue", "green"))</code> . Each set can contain different numbers of colors. Depending on the number of colors needed, the set with at least the number of required colors is used. The first color is always used for simple bar charts. |
| <code>base_gradient</code> | A color vector used for creating gradient fill colors, e.g. in stacked bar plots. |

Details

[Experimental]

Value

A theme function.

Examples

```
library(volker)
library(ggplot2)
data <- volker::chatgpt

theme_set(theme_vlkr(base_size=15, base_fill = list("red")))
plot_counts(data, sd_gender)
```

Index

* datasets

chatgpt, 2

chatgpt, 2

codebook, 3, 7

data_clean, 4, 5, 7, 11–13, 15–17

effect_counts, 4, 13, 15

effect_counts_items, 4

effect_counts_items_cor, 4

effect_counts_items_grouped, 4

effect_counts_one, 4

effect_counts_one_cor, 4

effect_counts_one_grouped, 4

effect_metrics, 5, 14

effect_metrics_items, 5

effect_metrics_items_cor, 5

effect_metrics_items_grouped, 5

effect_metrics_one, 5

effect_metrics_one_cor, 5

effect_metrics_one_grouped, 5

get_direction, 13, 15

html_report, 6

idx_add, 6

labs_apply, 7

labs_clear, 8

labs_restore, 9, 10

labs_store, 9, 10

plot_counts, 10, 12, 13

plot_counts_items, 10

plot_counts_items_cor, 11

plot_counts_items_grouped, 10

plot_counts_one, 10

plot_counts_one_cor, 11

plot_counts_one_grouped, 10

plot_metrics, 11, 14

plot_metrics_items, 11

plot_metrics_items_cor, 12

plot_metrics_items_grouped, 12

plot_metrics_one, 11

plot_metrics_one_cor, 12

plot_metrics_one_grouped, 11

report_counts, 12

report_metrics, 14

tab_counts, 12, 13, 15

tab_counts_items, 15

tab_counts_items_cor, 15

tab_counts_items_grouped, 15

tab_counts_one, 15

tab_counts_one_cor, 15

tab_counts_one_grouped, 15

tab_metrics, 14, 16

tab_metrics_items, 16

tab_metrics_items_cor, 16

tab_metrics_items_grouped, 16

tab_metrics_one, 16

tab_metrics_one_cor, 16

tab_metrics_one_grouped, 16

theme_vlkr, 17